Stavanger kommune

Azure Forvaltnings rammeverk

1. juli 2020

Dokumentinformasjon

	Navn	Rolle	Dato
Forfattere	Capgemini AS	Konsultasjon- og rådgivning	31.01.2019
Verifisert av	Capgemini Stavanger kommune		
Godkjent av	Stavanger kommune		

Revisjoner

Versjon	Dato	Endring	Endret Av
0.1	25.01.2019	Intern gjennomgang	Capgemini
0.8	28.01.2019	Første revisjon av Stavanger kommune	Capgemini / Stavanger kommune
0.9	29.01.2019	Andre revisjon	Capgemini / Stavanger kommune
1.0	01.02.2019	Overlevert til Stavanger kommune	
1.9	20.06.2020	Oppdatert arkitektur til sikker versjon. Oppdatert GDPR beskrivelse	Capgemini

Referanser til andre dokumenter

#	Dokument
1	Azure Data Catalog – rammeverk.docx
2	Hva er Personopplysninger og sensitive personopplysninger.doc
3	Styringsdokument Datasjø.docx
4	Prosjektmandat Innføring Datasjø.docx
5	Azure Data Factory.docx

Innhold

1 Introduksjon 4

1.1 Sammendrag 4

1.2 Avgrensinger og forutsetninger 5

1.3 Begreper og forkortelser 6

1.4 Prinsipper 8

2 Arkitektur 11

2.1 Løsningsarkitektur 11

2.2 Sikkerhetsarkitektur 14

2.3 Roller 18

2.4 Innebygd personvern og innebygd sikkerhet 20

3 Kildesystemer 24

3.1 Prinsipper 24

3.2 Kategorisering 24

3.3 Mål og leveranser i analysearbeid 26

4 Mottak 29

4.1 Systemer 30

4.2 Prinsipper 30

4.3 Sikkerhet 31

5 Lagring 31

5.1 Systemer 31

5.2 Prinsipper 31

5.3 Om datasjøen 31

5.4 Katalogstruktur 32

5.5 Aktive data 32

5.6 Inaktive data 32

5.7 Dokumentasjon av data i datasjø 33

6 Bruke data lagret i datasjø 33

6.1 Analyse og implementasjon av bruk av data i datasjø 34

7 Transformering 34

7.1 Prinsipper 35

7.2 Transformering 37

7.3 Overvåking og analyse av transformasjons-jobber 41

7.4 Sikkerhet 41

7.5 Last til Azure lagringstjenester 42

8 Betjening 42

8.1 Systemer 42

8.2 Definisjoner 43

8.3 Prinsipper 43
8.4 Sikkerhet 44
9 Visualisering/Presentasjonslag 44
9.1 Analyse 44
9.2 Applikasjoner/Tjenester 47
10 Overvåking 49
10.1 Rapporter og analyse rundt implementasjon av rammeverk 49
11 Brukerloggdatabase 50
12 Pseudonymisering 50
13 Hashicorp Vault – nøkkelhåndtering 50
14 Anonymisering 51
15 Datakatalog 51
15.1 Prinsipper 52
15.2 Roller og tilganger 52
15.3 Taksonomi og vokabular 52
15.4 Beste praksis i katalogisering av en kilde 52
16 Navnekonvensjoner 53
16.1 Ressursgrupper 53
16.2 Tjenester 54
16.3 Språk 54
17 Vedlegg A –Hva er data? 55
17.1 Informasjonsforvaltning / Informasjonsarkitektur 55
17.2 Klassifisering av kunnskap – DIKW pyramiden 56
18 Vedlegg B –Lover og Regler 57
18.1 Personopplysingsloven / GDPR 57
18.2 ISO og tilhørende referanser 60
19 Vedlegg C – Revisjon 61
19.1 Forvaltningsrammeverk 61
19.2 Sikkerhet 62

Azure forvaltningsrammeverk

1 Introduksjon

Stavanger kommunes strategiske mål er å få kontroll på og eierskap til egne data. Dette gjelder alle typer data fra kommunens ulike fagsystemer, samt andre typer data som for eksempel sensordata. Stavanger kommune har etablert en datasjø for å få alle data samlet i ett felles område. I tillegg til å gi kontroll og eierskap over dataene, vil datasjøen også gi verdi ved å legge til rette for analyse på og innsikt i de store og varierte datamengdene kommunen har.

For å tilrettelegge for denne type analyse og innsikt, har Stavanger kommune valgt Microsoft Azure dataplattformen med tilhørende komponenter. Løsningen vil åpne for håndtering av alle typer data, analyse på tvers av fagsystemer, bruk av data til andre og nye formål som for eksempel innsyn, smartbyutvikling og bedre tilgjengeliggjøring av åpne data.

1.1 Sammendrag

Dette dokumentet gir en oversikt over Stavanger kommunes regler og retningslinjer ved bruk av komponenter og tjenester på Microsoft Azure plattformen i form av et forvaltnings- og sikkerhetsrammeverk.

Rammeverket dekker hvordan man må gå fram for å analysere kilder, dokumentere dem og laste data opp i datasjøen, samt retningslinjer for hvordan data kan aggregeres, anonymiseres og ellers tilrettelegges for analyse og innsikt for sluttbrukere og mottagende systemer på en sikker måte som respekterer og tar vare på personvernet.

Sikkerhet ved bruk av de ulike komponentene er en viktig del av rammeverket.

Dette dokumentet er i utgangspunktet rettet mot personer med teknisk innsikt og som i kraft av sin rolle enten skal arbeide med datasjøen eller skal forvalte rammeverket. Under følger et forenklet sammendrag som flere kan ha nytte av.

Flyt av data i en datasjø går som regel igjennom 6 steg eller stadier fra kildesystemer fram til konsum hos mottager. Samtidig er det ofte gunstig å se på dataflyten som bevegelser i separate lag, og dette rammeverket knytter tjenestene som skal brukes i datasjøen til lag. Sikkerheten i datasjøen er koblet til de ulike tjenestene i de spesifikke lagene de skal benyttes. Enkelte tjenester vil ha en rolle i flere av lagene.

I dette dokumentet ser vi på datasjøen utfra tre bruksområder:

- **Del 1 – Hente data fra kilde og lagre i datasjø**
    - Kildesystemer – Analyse av kildesystem for å hvordan håndtere kopiering av data inn til datasjø, dokumentasjon og videre bruk av data til ulike formål
    - Mottak - Tjenester som fungerer som inntak til datasjøen
    - Lagring - Hvordan lagre rådata i datasjø

  Data hentes fra kildesystem uten å endres og lagres dermed som rådata i datasjø.
  Gode rutiner for analyse av av kildesystemer er viktig for at rett sikkerhetsnivå kan legges for data fra de ulike kildene og personopplysninger kan håndteres på en riktig måte. Forvaltning av tilgang til rådata, prinsipper og rutiner for loggføring av bruk av dataen sikrer sporbarhet og kontroll.

- **Del 2 – Bruke data lagret i datasjø**
    - Transformering – Hvordan lese og transformere data fra datasjø på en sikker måte
    - Betjening – Hvordan lagre data på best og tryggest for de ulike behovene/formålene
    - Presentasjon – Presentasjon/Visualisering av data til mottakere

Transformering av data er en operasjon som både kan innebære pseudonymisering, anonymisering av sensitive data, aggregering for å levere data til bruk for generell analyse eller sammenstille data fra ulike systemer slik at det kan gjøres analyse på tvers uten behov for dataforskningskompetanse.

Betjening er å gjøre dataene tilgjengelig for konsum av ulike mottagere – det være seg andre datasystemer eller sluttbrukere. Tilgangskontroll og andre sikkerhetshensyn spiller inn på hvilke tjenester som vil bli valgt til betjening avhengig av type data.

- **Del 3 – Felles Funksjoner**
    - Datakatalog – Dokumentasjon av alle data i de ulike lagringsmediene
    - Etterlevelse – Tjenestene som sikrer at kommunen kan etterleve personvernlovgivingen
    - Overvåking – Hvordan overvåke de ulike tjenestene i plattformen

Datakatalogen er nøkkelen til datasjøen, her vil brukere kunne søke etter relevante data for å drive selvbetjent analyse. Sammen med datakatalogen trengs noen tilleggstjenester for å sikre at kommunen har full kontroll på hvilke data man har om den enkelte innbyggeren, hvor de ligger, hvordan de er brukt og av hvem.

Etablere å etterfølge rutiner og verktøy for å overvåke og følge opp systemet når det er i bruk er like viktig som å bygge god sikkerhet inn i datasjøens arkitektur.

1.2 Avgrensinger og forutsetninger

1.2.1 Avgrensinger

Prosjekter, prosjektplan, prosjektledelse og prosjektgjennomføring for utvikling av kilde, datasjø, analyse med mer. Det vil være naturlig at flere elementer fra rammeverket vil inngå som oppgaver og del-leveranser i et prosjekt. Videre vil man også finne ressurser i form av roller som vil være nødvendige for å gjennomføre prosjekter. Noen av prinsippene vil bli direkte milepæler i et prosjekt. Et eksempel her er godkjenning av kopiering av data til datasjø.

Informasjonsforvaltningsrammeverk er et eget fagfelt med omfattende regler for forvaltning av informasjon. Analyse av kilde, lagring av data og bruk til analyse vil bli direkte berørt av regler som naturlig vil bli dekket i et slikt rammeverk. Noen eksempler er masterdata, datakvalitet, dataenes livssykluser, godkjenning av tilganger til data osv. Leveranse der dette rammeverket vil bli brukt vil måtte knyttes tett sammen med informasjonsforvaltningsrammeverket.

Brukerstøtte og support er ikke dekket. Med brukerstøtte og support menes innmelding av saker fra brukere på feil og feil rapportert fra løsning. Det antas at Stavanger kommunes eksisterende brukerstøtte-organisasjon håndterer dette. Det som er viktig er at dette må utvides til å støtte eksisterende og framtidige løsninger levert i Azure plattformen.

Rammeverket dekker ikke behov til **opplæring** av ressurser som vil være direkte eller indikert involvert i leveranser eller bruk av data og tjenester.

Rammeverket vil fokusere sterkt på sikkerhet og regler rundt implementasjon. Rammeverket dekker ikke selve prosessen med å få **tilgang til data**. Stavanger kommune har allerede en prosess på dette, og den må utvides for å støtte denne løsningen. Videre er det prosesser for å få **lov til å bruke data** fra datasjø. Denne evaluering og godkjenning faller utenfor dette rammeverket. Rammeverket hjelper til å skissere hvordan data skal flyte og deles.

Rammeverket dekker i skrivende stund – versjon 1.9 – ikke mer avansert **analysetjenester** som maskinlæring (Machine Learning/ Deep Learning), kunstig intelligens (Artificial Intelligence) med mer. Det vil derimot være naturlig for Stavanger kommune å utvide rammeverket med et analyselag som tar for seg slike tjenester skulle det være fremtidig behov for dette.

Råd og veiledning i forbindelse med **utvikling av applikasjoner og tjenester** er ikke dekket. Råd og regler for bruk av data fra datasjø vil måtte bli tatt hensyn til i slike implementasjoner.

Rammeverket dekker heller ikke hvordan innbygger skal kunne interagere med datasjøen for å kunne utøve sine retter i henhold til personvernlovgivningen.

Beskrivelser og sikkerhetskomponenter i dette dokumentet skal ikke forstås som en løsning for hvordan å **unngå angrep eller cybertrusler.** Det poengteres at enhver skybasertplattform kan utsettes for angrep og tilsiktede handlinger, dersom angriper har de nødvendige ressurser tilgjengelig. Derfor er det å fokusere på risikominimerende tiltak viktigere. Risikoarbeidet og øvelser skal sys inn i implementasjon og forvaltning av datasjøen. Dette rammeverket må forstås som et levende dokument som **krever videre og kontinuerlig arbeid**. Dokumentet må **evalueres og revideres** på bakgrunn av bl.a. endringer i lover og regler, den teknologiske utviklingen, endringer i beste praksis samt erfaring/lærdom fra lignende prosjekt. Les mer om dette i *Vedlegg C – Revisjon*

Forvaltningsrammeverk.

1.2.2 Forutsetninger

Stavanger kommune har et ansvar og en plikt som dataeier og databehandler til å ha en oversikt over alle regulativ og lover som vil gi føringer for bruk og behandling av data. I senere seksjoner vil dette blant annet dekke flytting og kopiering av data til datasjø, og bruk av data i datasjø. Eksempler på begreper er behandling av personopplysninger, formål med databehandling, tjenstlige behov og prosess for eventuell godkjenning fra Datatilsynet.

Det forventes at leseren av dette rammeverket har noe teknologisk kunnskap hva gjelder ulike komponenter i Azure og lignende. Som bruker av rammeverket forventes det utvidet teknisk forståelse.

1.3 Begreper og forkortelser

Begreper	Forklaring	Engelsk
Stavanger kommune	Forvaltningsenheten Stavanger kommune	
Stavanger kommunes ansatte	Ansatte i Stavanger kommune som har en rolle i forhold til databehandling	
On-premises	Systemer/Løsninger som er installert og kjører på infrastruktur (serverer) som er interne i Stavanger kommune. I tillegg vil også deler eller hele forvaltningen være organisert av Stavanger kommune. Dette i motsetning til systemer og løsninger som kjører eksternt eller i skyen.	On-premises
Skytjeneste	Skytjenester er en samlebetegnelse på alt fra dataprosessering og datalagring til programvare på servere som er tilgjengelig fra eksterne serverparker tilknyttet internett.	Cloud Computing
Azure	Azure er en offentlig skytjeneste levert av Microsoft.	Azure
Datakatalog	En datakatalog er som navnet sier oppslagsverk der alle datakilder dokumenteres med metadata og kategoriseres slik at det er enkelt for brukere å søke etter data på tvers av flere systemer.	Data Catalog
Datasjø	Lagringsområde for data på servere tilgjengelig via internett. Kan inneholde både strukturerte data (databaser) og ustrukturerte data (rådata).	Data Lake
Data Livsløpsadministrasjon	Få kontroll på hvor mye data som lagres, hva som er lagret og hvor det ligger. Og ikke minst hvor lenge man trenger å lagre	Data Lifecycle Management

	dataene. En riktig DLA tilnærming vil sørge for at man ikke betaler for data som ikke brukes eller trengs å tas vare på.	
Bruker/Sluttbruker	Unike personer/individer som bruker tjenester	User End User
Mottaker	En mottaker kan være både en bruker (menneske), offentlig-/privat institusjon som kan mottak data. Disse kan være både interne og eksterne.	Recipient
Kilde (Datakilde)	Alle mulige datakilder. Det kan være en applikasjon, rapport, database, datasett osv.	Data Source
Kildesystemer	Alle systemer som blir brukt i nye Stavanger kommune (inkludert Rennesøy- og Finnøy kommune) innenfor de ulike tjenesteområdene. Vil alltid være MASTER	Source Systems
Kryptering	Som i eks "256-bit AES kryptering". Ulike krypteringsmetoder bruker noe ulike kodingsteknologier, men felles er at kun mottaker med korrekt nøkkel kan lese de kodede dataene.	AES: Advanced Encryption Standard
Database (DB)	Relasjonsdatabase – system for å lage strukturerte data i rader, delt opp i kolonner og plassert i ulike tabeller etter type innhold. Relasjonene mellom tabellene settes ved hjelp av nøkler, der hver tabell har sitt nøkkelsett og fremmednøkler i den enkelte tabellen viser til nøkkelsett i en relatert tabell.	(Relational) database
BLOB	Et lagringsformat for ustrukturerte data av alle typer i Microsoft Azure. BLOB-lagring er et rimelig lagringsalternativ for data man ikke bruker så ofte (kalde data).	Binary Large OBject
Strukturerte data	Data lagret i databaser eller andre applikasjoner som strukturerer data	
Ustrukturerte data	Data med ingen eller liten grad av struktur. Kan f.eks. være tekstdokumenter, legejournaler, binærstrømmer eller sensor data fra IOT.	
Autentisering	Autentisering er prosessen med å verifisere autentisiteten av en entitet. En entitet kan være en person, en gruppe eller et annet datasystem.	Authentication
Autorisering	Autorisasjon er prosessen med å bestemme om en autentisert entitet er berettiget en tilgang.	Authorization
Selvbetjent analyse	Å gjøre data og verktøy tilgjengelig for ikke IT-folk slik at de kan selv utnytte data til å ta bedre beslutninger.	Self-service
SDK	Utviklingsverktøy mot en spesifikk programvarepakke, i dette dokumentet for Azure (Software Development Kit).	SDK
GDPR	EUs forordning for personvern i databehandling, som også Norge har sluttet seg til.	General Data Protection Regulation

Personopplysninger	Alle typer informasjon som direkte eller indirekte kan identifisere en fysisk person som er i live.	Personally identifiable information (PII)
Pseudonymisering	Pseudonymisering er en prosedyre for databehandling og avidentifikasjon der personlig identifiserbare informasjonsfelt i et dataregister erstattes av en eller flere kunstige identifikatorer, eller pseudonymer	Pseudonymization
Pseudo ID	En person identitet som er pseudonymisert	Pseudo ID
Behandlingsgrunnlag	All behandling av personopplysninger må ha et rettslig grunnlag (lovhjemmel) for å være lov. Det betyr at virksomheten på forhånd må ha identifisert om det finnes et behandlingsgrunnlag. Hvis ikke det finnes, er behandlingen ulovlig.	Lawful basis for processing
Overvåke	Systematisk og regelbundet innsamling og vurdering av data, i dette tilfellet av Azure komponenter.	Monitoring
Logg	En datafil som i kronologisk rekkefølge lagrer relevant hendelser.	Log
Personvernkonsekvensvurdering / DPIA	Vurdering av personvernkonsekvenser, i henhold til personopplysningsloven.	Data Protection Impact Assessment
ROS-analyse	Risiko- og sårbarhetsanalyse.	Risk and vulnerability analysis
ISO 27001	International Organization for Standardization. 27000-familien omhandler informasjonssikkerhet.	ISO 27001
Anonymiserte data	Data som ikke inneholder informasjon om, og ikke kan spores tilbake til bestemte personer.	Anonymized data
Ordliste (Vokabularer)	Stavanger kommune sin liste av ord og deres forklaring (og bruk). Det finnes også flere standard ordlister på ulike språk og fagområder.	Glossary (Business Glossary)
Taksonomi	Klassifisering av informasjon.	Taxonomy
IOT	Tingenes internett. Alle typer gjenstander som kan koble seg til internett, og er tilgjengelig via nettet.	Internet of Things

1.4 Prinsipper

Følgende hovedprinsipper ligger til grunn i valg av arkitektur og sikkerhetsrammeverk. Seksjonene senere i rammeverket vil ha prinsipper som bygger videre på disse overordnede prinsippene.

1.4.1 Sikkerhetsprinsipper

Prinsipp	**1 Behandlingsansvarlig for kildesystem skal godkjenne all bruk av ikke-anonymiserte data**		
Beskrivelse	All bruk av ikke-anonymiserte data skal være godkjent av kildesystemets dataeier, og det skal foreligge et lovlig behandlingsgrunnlag.	**Betydning**	Ikke-anonymiserte data vil bare kunne aksesseres når godkjenning fra kildesystemets dataeier foreligger. Det skal alltid foreligge et lovlig behandlingsgrunnlag for all bruk av persondata.
Prinsipp	**2 Data kan kun lagres hvis det finnes lovhjemmel for dette.**		
Beskrivelse	Skal data legges i datasjø må det finnes en lovhjemmel for dette.	**Betydning**	Det er viktig at man ikke legger opp data i datasjøen som ikke kommunen har lov eller

			legitimt grunnlag for å til å lagre i datasjøen.
Prinsipp	**3 Data kan ikke brukes til andre formål enn hva det er samlet inn til**		
Beskrivelse	Skal innsamlede data brukes til andre formål må det foreligge et behandlingsgrunnlag som tillater dette.	**Betydning**	Det skal alltid foreligge et behandlingsgrunnlag for bruk av data i datasjøen. Skal data benyttes til andre formål enn hva det er samlet inn til i kildesystemet må det avklares at behandlingsgrunnlaget eller andre forhold ikke hindrer bruken av dataene til dette nye formålet.
Prinsipp	**3 Data som inneholder sensitive opplysninger kan anonymiseres for å tilrettelegge for videre bruk**		
Beskrivelse	Ved å anonymisere sensitive data i et datasett kan datasettet gjøres tilgjengelig for konsumenter som ellers ikke hadde hatt tilgang. Slik kan sensitive data kan gi stor verdi uten å gå på bekostning av personvernet.	**Betydning**	Sensitive data vil ikke bli tilgjengelig for analyse. Persondata vil bli anonymisert og/eller aggregert bort.
Prinsipp	**4 Behandlingsansvarlig for kildesystem skal godkjenne opplasting av data til datasjøen**		
Beskrivelse	Kildesystemets behandlingsansvarlig må akseptere risikoen forbundet med å dele data fra opprinnelig system med datasjø.	**Betydning**	Dersom sensitive eller personopplysninger lastes opp til datasjøen skal kildesystemets behandlingsansvarlige akseptere gjenværende risiko som angitt i risiko- og sårbarhetsanalysen for datasjøen.
Prinsipp	**5 Ved konsumering og analyse av ikke-sensitive og/eller anonymiserte data er det behandlingsansvarlig for datasjøen som eier risikoen ved evt. sammenstilling og deling av data**		
Beskrivelse	Sammenstilling av data kan utilsiktet gi en utenforstående innsikt i forhold som det ikke er et tjenstlig behov for. Det er behandlingsansvarlig for datasjøen som eier risikoen ved dette og må foreta den endelige vurderingen før tilgang til data tillates.	**Betydning**	Godkjenningsprosesser må være på plass for all type data som deles.
Prinsipp	**6 Ved all bruk av data i datasjøen skal behandlingsgrunnlaget være innhentet/avklart**		
Beskrivelse	Ved konsumering og analyse av data er det bestiller som har ansvar for at behandlingsgrunnlaget til data er innhentet/avklart i henhold til gjeldende nasjonale regelverk.	**Betydning**	Tilgang til data gis først når dette er dokumentert.
Prinsipp	**7 Bestiller har ansvar for at det utføres en risikovurdering før konsumering og analyse av data i datasjøen.**		
Beskrivelse	Risikovurdering og/eller sikkerhetsvurdering skal alltid foreligge i forkant av presentasjon av data som blir hentet fra datasjø og gjort tilgjengelig til mottakere.	**Betydning**	Bestiller av data må ikke bare innhente godkjenning av bruk av data, men også sikre at implementasjonen og presentasjonen av data til mottakere, blir kontrollert og sjekket. Dette kan være før første produksjonssetting, men også ved gitte intervaller etterpå.

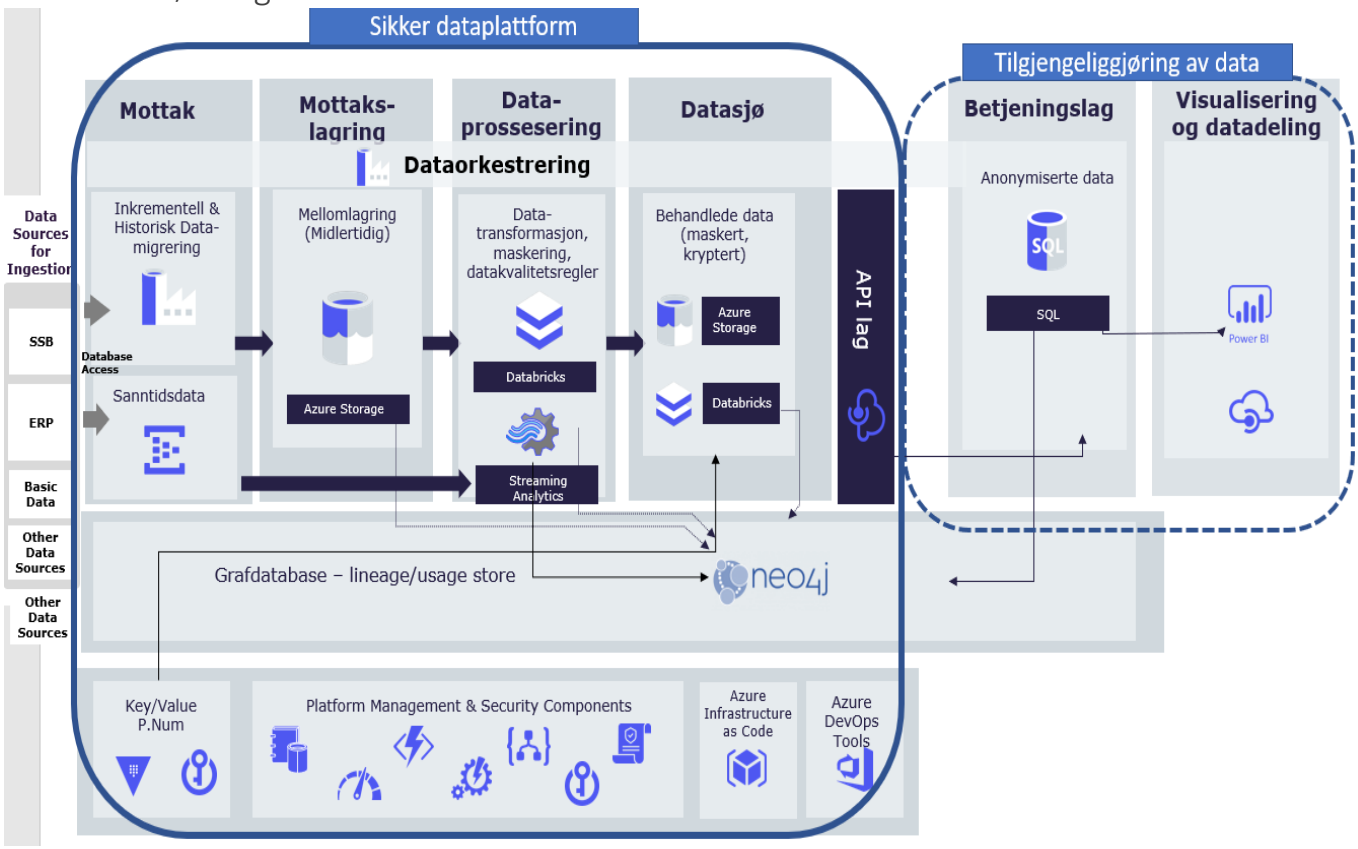
Andre prinsipper

Prinsipp	**8 Bruk av tjenester og komponenter i Azure skal være satt opp så kostnadseffektivt som mulig**		
Beskrivelse	Nye tjenester og komponenter i Azure skal ikke settes opp uten at det har blitt gjort en kost/nytte analyse og skal settes opp med laveste dimensjonering analysen har indikert.	**Betydning**	Selv om mange tjenester i Azure kan skaleres opp og så ned igjen, så er det også flere tjenester som kun kan skaleres opp. Det å overdimensjonere tjenester er unødvendig bruk av penger.
Prinsipp	**9 Forvaltningsrammeverkets regler og veiledninger skal kunne flyttes/videreføres på andre plattformer**		
Beskrivelse	Forvaltningsrammeverkets regler og veiledninger skal i så stor grad som mulig være universelle for andre plattformer slik at en eventuell flytting/videreføring på annen plattform lar seg gjøre.	**Betydning**	Regler og veiledninger skal ikke være så knyttet til spesifikke tekniske plattformer at de ikke kan gjenbrukes på generell basis.
Prinsipp	**10 Åpne data skal tilgjengelig gjøres for Stavanger kommune sin åpne plattform**		
Beskrivelse	I henhold til Stavanger kommune sin strategi skal alle data som er definert som åpne gjøres tilgjengelig. Det må foreligge en tydelig kategorisering av data og godkjenning for at data skal falle i denne kategorien.	**Betydning**	Data som hentes over i datasjø og blir kategorisert som åpne, skal gjøres tilgjengelig for Stavanger kommunes åpne dataplattform https://open.stavanger.kommune.no/. Denne plattformen vil måtte hente data fra betjeningslaget.
Prinsipp	**11 Kildesystem vil alltid være master for data, mens datasjøen vil inneholde en kopi**		
Beskrivelse	Datasjøen henter alltid data fra en kilde. Datasjøen forholder seg til at den opprinnelige datakilden alltid er den mest **riktige** kilden for data.	**Betydning**	Dersom data i datasjøen og opprinnelig datasystem avviker, vil opprinnelig datasystem være å betrakte som riktig.
Prinsipp	**12 Datasjøen skal ikke benyttes som grunnlag for arkiv**		
Beskrivelse	Datasjøens intensjon er å samle, sammenstille og anvende data. Data som har historisk betydning for oppbevaring har en annen betydning. Arkivverdig data bør derfor forholde seg til andre plattformer enn datasjø, men kan fungere som en datakilde inn til datasjøen.	**Betydning**	Arkivverdig data skal oppbevares i arkivsystem og forvaltes etter egne regler.
Prinsipp	**13 All anvendelse av tjenester fra datasjøen skal være kontraktsfestet med mottakere**		
Beskrivelse	Alle mottakere av data fra datasjøen (eksterne som interne, offentlige og private), må akseptere en kontrakt før bruk, som regulerer ansvaret og bruken av tjenesten.	**Betydning**	Dersom en 3. part misbruker informasjon og data fra datasjøen til et annet formål enn det som er kontrakts bestemt, er mottaker er selv ansvarlig for misbruket med tilhørende konsekvenser. Mottakere har også plikt til å rapportere hvis man får tilgang til data som man normalt ikke vill hatt.
Prinsipp	**14 Datasjøen skal teknisk være i stand til å kunne hente eller motta data fra alle systemer i Stavanger kommune**		
Beskrivelse	Datasjøen bør inneha de verktøy som trengs for å kunne hente data fra samtlige datasystemer i kommunen.	**Betydning**	Uthenting av data er kritisk faktor for datasjø-konseptet. Ettersom datasystemene er vurdert å være master, bør prinsippet

	Behandlingsansvarlige bør være bevisst på at data skal kunne hentes ut fra datasystemer til datasjø ved behov.		følges av både behandlingsansvarlig for Datasjø og behandlingsansvarlige av opprinnelige datakilder.

2 Arkitektur

2.1 Løsningsarkitektur



Å flytte fra en intern dataplattform og ut i skyen kan være en komplisert affære, da skyleverandører som Microsoft ofte har en rekke tjenester som tilsynelatende kan brukes om hverandre.

I dette kapittelet går vi gjennom Stavanger kommunes løsnings- og sikkerhetsarkitektur og de ulike komponentene Stavanger kommune har valgt å bruke i sin datasjø.

Flyt av data i en datasjø går som regel igjennom 6 steg. Men i en løsningsarkitektur blir de 6 stegene til tjenestelag. Så når vi omtaler disse 6 stegene i arkitekturen og påfølgende utdypende kapitler så vil vi bruke begrepet lag.

2.1.1 Beskrivelse

Løsningsarkitekturen følger Microsoft beste praksis. For hvert av de 6 lagene i datasjøen er det flere ulike komponenter som kan brukes, men hver komponent har sin spesifikke funksjon. Komponenter kan være representert i mer enn et lag. De må da følge de regler for det respektive laget slik det er presentert i matrisen under.

2.1.1.1 Mottak:

	**Azure Data Factory (ADF)** er en tjeneste for å definere, orkestrere og administrere flytting samt transformasjon av data opp til skyen. ADF kan jobbe mot forskjellige kilder, inkludert on-premises SQL Server og andre relasjonsdatabaser, flatfiler m.m. og egner seg best for batch opplasting av data i gitte intervaller og for en del transformasjon via Mapping Dataflows eller altså ved å orkestrere transformasjon via Databricks eller Functions. Stavanger kommune bruker ADF til å orkestrere alle mottaksjobber og for å orkestrere dataflyten og komponentene i de ulike stegene i datasjøen
	**Azure Event Hub** (herunder også IoT Hub) er et mottakspunkt for telemetri, og er optimalisert for å kunne ta imot store mengder data samtidig uten å bli overbelastet. Mottak direkte i en database, eller via et API, kan kreve prosessering som gjør at det er mulig å sende inn mer data enn systemet klarer å sortere samtidig. Event Hub sørger for at selve mottaket ikke er et problem, og at data ikke går tapt i dette leddet. Det er derfor en viktig komponent i mottaksfasen i datadrevne løsninger. IoT Hub skal kun brukes i rene IoT scenarier der det er behov for autentisering per enhet/sensor, der det er behov for toveis kommunikasjon mellom sky og enhet/sensor, der det er behov for analyse nære enhetene i felt (kant-behandling/«edge computing») og eventuelt hvis avleverende system kun støtter protokollen Azure IoT Hub alene støtter (SMQP). Ellers benyttes Azure Event hub for alle telemetridata.
	**Azure Functions** er en serverløs tjeneste som kan designes til en hel rekke formål, som henting av data fra FTP eller API-tjenester. For å overvåke filområder der avleverende system ikke sender data til faste intervall eller tider. Ettersom tjenesten er avhengig av at det skrives spesialkode, må Stavanger kommune etablere standarder for sikker koding på alle instanser av tjenesten. Azure Functions er et supplement til de to overnevnte tjenestene der de mangler funksjonalitet.
	**Azure Logic Apps** er en serverløs tjeneste for å definere, orkestrere og automatisere ulike arbeidsflyter som for eksempel flytting og transformasjon av data. Logic Apps og Data Factory er komplementære tjenester som kan benyttes sammen og kan kalle hverandre. Logic Apps er den tjenesten som virker best sammen med API'er, og brukes for å orkestrere arbeidsflyter som kan eller skal trigges av API-kall.

2.1.1.2 Lagring:

	**Azure Data Lake Store Gen. 2**(ADLS) er konseptuelt tenkt som en ubegrenset harddisk. Merk at tjenesten har nå fått nytt navn – Azure Storage account with hierachical namespace og velges først i avansert stadie i opprettelse av lagringstjeneste. Tjenesten er filbasert, og støtter kataloger/mapper for å strukturere innholdet. ADLS støtter også adgangskontroll etter RBAC-prinsippet på filnivå, i motsetning til BLOB Store som kun har delt tilgangsnøkkelkontroll. Grensesnittet er WebHDFS og støtter dermed en rekke tjenester utenfor Azure-sfæren i tillegg til Azure Databricks. Stavanger kommune benytter som beskrevet senere i dette dokumentet Databricks til transformasjon. Det er ingen begrensninger på antall filer, størrelse eller format. Dette gjør at det er veldig fleksibelt for store datamengder enten det er binærfiler eller ulike tekstbaserte formater. Slik sett ikke direkte sammenlignbart med en database, men vil i mange løsninger fungere i samspill med databaser. Dette kan være både lagring av rådata som etterpå havner inn i en database, eller at det er lettere å bearbeide uttrekk fra en database når dataene er formatert som filer. ADLS kommer også med ulike pris- og funksjonalitetsnivå : Varm, Kald og Arkiv der man betaler for lagring, antall skrive og leseoperasjoner og responstid. Dette understøtter Data Livsløpsadministrasjon – og utbedres fortløpende.
	**Azure Blob Store** er en lagringstjeneste med flat struktur, altså ingen mappestruktur. Tjenesten har fått nytt navn – Azure Storage account. Den har begrensninger på filstørrelse og filtype, selv om dette er mer en teoretisk begrensning. Tjenesten er beregnet for ren lagring av data og har begrensninger på skrive/leseoperasjoner som gjør den veldig lite egnet til analytisk behandling av data i tjenesten. Er standard tjeneste valgt når man lager en storage account. Støtter versjonering av filer og angring av sletting.

	**Azure SQL Database** er en relasjonsdatabase (MS SQL Server). Dette er en relativt kostbar tjeneste da den kombinerer både lagring og transformering/beregning. Det kan i visse tilfeller være kostnadsnyttig å bruke en SQL DB til lagring, men oftest vil denne tjenesten kun benyttes i betjeningslaget.

2.1.1.3 Transformering:

	Azure **Databricks** er en containerbasert SPARK løsning. Spark er den minnebaserte analytiske motoren til Hadoop. Hadoop er de-facto standard for stordataprosessering, og en rekke tester har vist at Spark er den raskeste motoren for analyse på store datamengder. Databricks gjør kraften i Spark tilgjengelig i Azure på en kostnadseffektiv og enkel måte. Databricks kan brukes til alt fra enkel transformering, sanntidsanalyse til avansert maskinlæring.
	**Stream Analytics** er en sanntids analysetjeneste som kan gjøre spørringer på data etter som de kommer inn. Dette vil gjerne fungere som et første ETL-steg, før en mer detaljert analyse gjøres. Stream Analytics benytter et begrenset utvalg av T-SQL språket for sine spørringer. Men kan samtidig utvides med maskinlæringsalgoritmer og andre spesialtilpasninger.

2.1.1.4 Betjening:

	**Azure SQL Database** Er en standard relasjonsdatabase i Azure. Tilsvarende i funksjonalitet til SQL Server, men alltid siste versjon og skalerbar, uten å måtte bruke penger på oppgradering av server.
	**Azure SQL Pools (Synapse Analytics)** Er SQL-basert lagring med mulighet for massiv skalering. Gjennom PolyBase kan SQL Pools kombinere data fra både relasjonsdatabaser, datasjø og andre kilder direkte i datavarehuset. SQL Pools benytter T-SQL som er kjent fra standard MS SQL Server. Det er ferdige integrasjoner mot Power BI, Azure Data Lake, Machine Learning, med mer.
	**Cosmos DB** Er en distribuert multi-modell databasetjeneste. Den kan benyttes som dokumentdatabase, eller som et alterntativ til Neo4J som grafdatabase – spesielt knyttet til brukerloggdatabasen.
	**Neo4J** Er en grafdatabase for avansert analyse. Grafdatabaser er også gode til å holde kontroll på avanserte relasjoner mellom data, slik som for eksempel hvilke data som ligger hvor, om hvem og hvor disse dataene er brukt. I dette rammeverket er Neo4j brukt til brukerloggdatabase, for å kunne levere effektivt på personvernlovgivningens krav om datasubjektets rettigheter. Men det finnes også en rekke andre områder der kommunen kan dra nytte av graf kapabiliteter senere.
	**Azure Datalake Store** Denne tjenesten er den samme som under «Lagring», men det er ikke uvanlig at man benytter datalake store også til å lagre ferdig transformerte data.

2.1.1.5 Presentasjon:

	**Power BI** er Microsoft sitt visualiseringsverktøy i skyen, og passer for de aller fleste analyse og rapporterings formål.

	Tjenester og applikasjoner bruker typisk **API** og **Webservice** Azure støtter standard REST API og har integrert API i et felles grensesnitt – API Gateway som også bidrar med muligheter til sterk tilgangskontroll samt å sette opp bestillinger på levering av data. API'er er brukt i referansearkitekturen i dette rammeverket for å gjøre anonymiserte data tilgjengelig for rapportering, og for å kunne ha en sikker interaksjon med eksterne tjenester – både eksisterende og fremtidige – som for eksempel fremtidig Innbyggerportal.

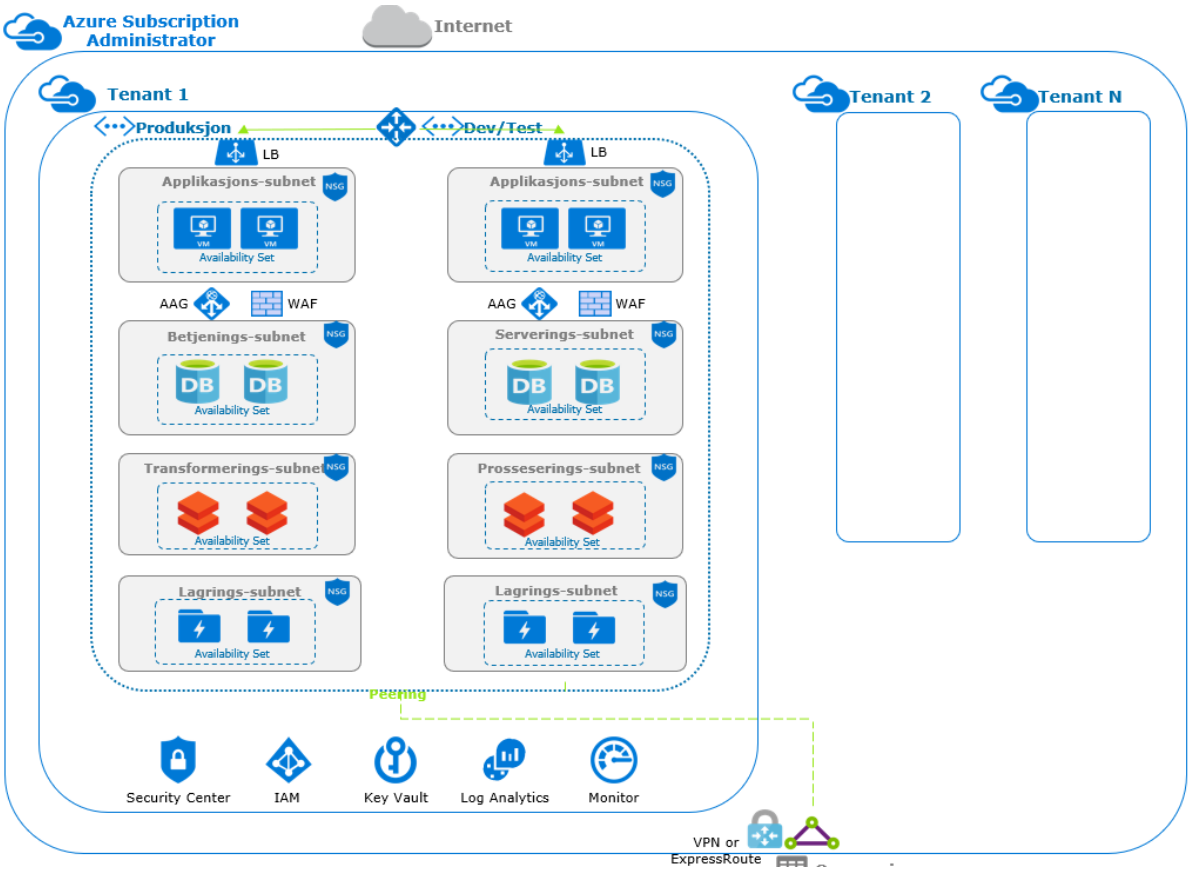
2.1.1.6 Felles funksjoner:

	**Data Catalog** Azure Data Catalog fungerer som en katalogtjeneste med oversikt over datakilder i en organisasjon, samt metadataene knyttet til disse kildene. Intensjonen er at den skal gi en enkel oversikt og veiledning for de som analyserer og konsumerer data i en bedrift.
	**Azure DevOps + Git** Azure DevOps er et system for å sørge for kilde- og versjonskontroll av all kode, og knytte koden opp mot et oppgavesystem som baserer seg på smidig utviklingsmetodikk for å sette opp, tildele og utføre oppgavene. Til dette ligger også muligheten til å automatisere deler av prosessen med utrulling av endringer og ny funksjonalitet i en Continuous Integration/Continuous Deployment loop. Dette tilsvarer det som gjerne kalles DevOps – og i datasjøsammenheng DataOps.

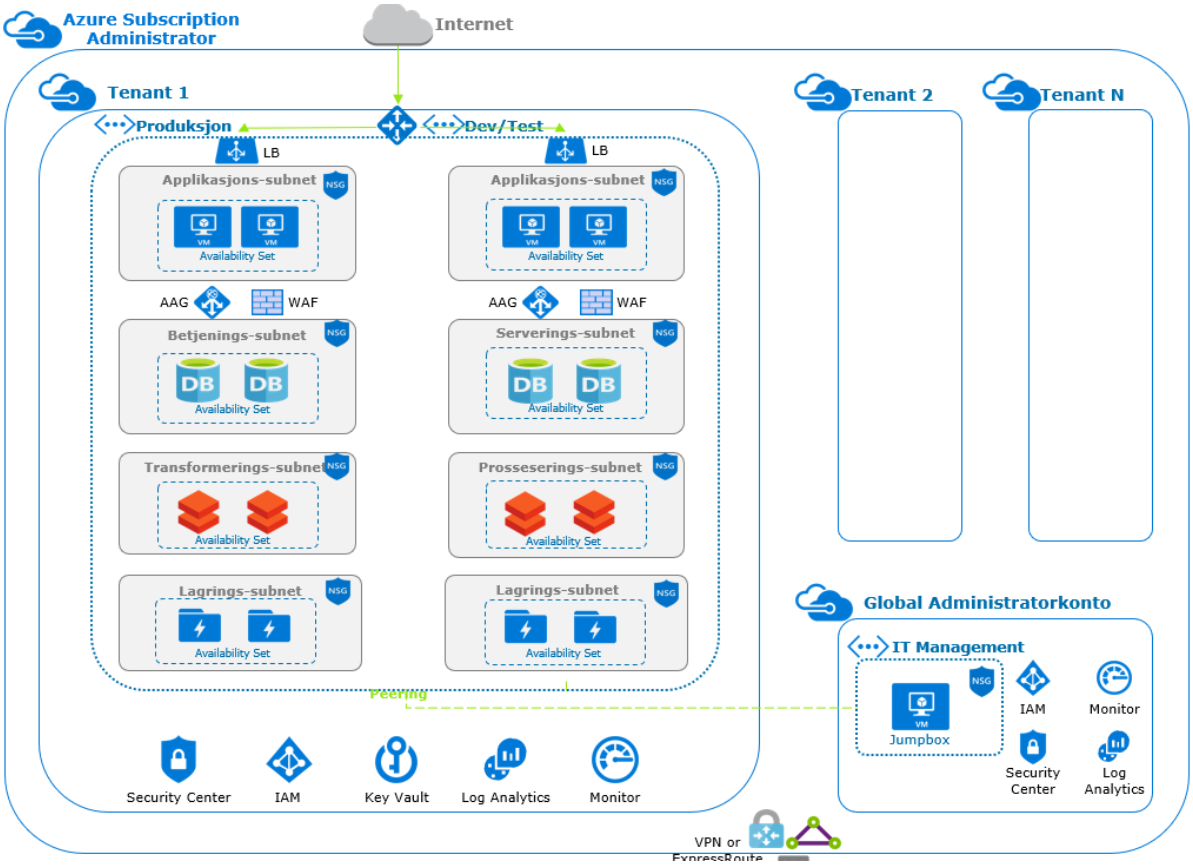
2.2 Sikkerhetsarkitektur

Sikring av data i en skytjeneste er like viktig som sikring lokalt. Det finnes en hel rekke verktøy i Azure for å øke sikkerheten, og her følger en beskrivelse av hvordan de benyttes i Stavanger kommunes sikrede datasjø. Alt er satt opp etter beste praksis og med de verktøy Microsoft Azure tilbyr. De 6 lagene i datasjøen kan gjenkjennes i skissen under, om enn i et annet oppsett enn i selve løsningsarkitekturen. Det er på tidspunktet dette dokumentet er laget ikke aktuelt med en direkte åpning mot internett, og skisse 1 viser anbefalt arkitektur – der det ikke er nødvendig med ekstra sikkerhetsmekanismer. Skisse 2 viser arkitekturen slik den anbefales lagt opp for å håndtere dette hvis behovet dukker opp.

Skisse 1. Arkitektur uten tilgang fra internett



Skisse 2. Arkitektur når det skal gis tilgang fra internett.

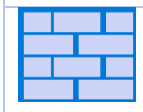


2.2.1 Prinsipper i sikkerhetsarkitektur

Prinsipp	**Lagdeling**		
Beskrivelse	Alle steg i sjøen legges i egne lag, med eget sub-nett.	**Betydning**	Ved å avgrense de ulike stegene i egne sub-nett så kan man innføre strengere tilgangskontroll.
Prinsipp	**Tilgangsgrupper**		
Beskrivelse	Hvert sub-nett har en egen tilgangsgruppe, Network Security Group(NSG).	**Betydning**	Ved å avskjerme hvert sub-nett med egen tilgangsgruppe skaper vi en adgangskontroll som har høy granularitet.
Prinsipp	**Mottakere**		
Beskrivelse	Mottakere har kun tilgang til betjeningslaget. Kun maskinkontoer har tilgang til de underliggende lag. Og disse har kun tilgang til laget rett under seg, og eventuelt laget rett over.	**Betydning**	For å sikre at ingen skaffer seg tilgang til data som ikke er klare til visning, skal kun maskinkonti ha tilgang til lagene under Visualisering. Maskinkonti skal kun ha tilgang til laget rett under, eventuelt også til laget direkte over.
Prinsipp	**Miljøer**		
Beskrivelse	Det skal være separate miljø for produksjon og utvikling.	**Betydning**	Ved å skille mellom miljøer – og data i miljøene – kan utviklere få tilgang til de nødvendige komponentene i lag som de vil være låst ute fra i produksjonsmiljøet.
Prinsipp	**Passord**		
Beskrivelse	Stavanger kommunes egen passordpolicy styrer kompleksiteten av brukeres passord. Men maskinkontoer bør ha strengere sikkerhet.	**Betydning**	Man kan ikke forvente at brukere har alt for komplekse passord, og de færreste passordregler krever dette. Maskinkonti og andre konti som trenger tilgang, bør ha komplekse passord. Disse skal lagres i Azure Key Vault.
Prinsipp	**Logging og monitorering**		
Beskrivelse	Alle tjenester skal ha detaljert logging aktivert, og alle logger skal overvåkes og analyseres.	**Betydning**	Logging av alle tjenester og overvåking av loggene er standard. Administratorer har ansvar for oppfølging av overvåkingen.

2.2.2 Dette er de ulike sikkerhetskomponentene som er i bruk:

	**Azure Virtual Network** Virtual Network tillater segmentering og isolering av ressurser. Innen det enkelte virtuelle nettverket settes opp egne sub-nett som er komponentene for segmentering.
	**Azure Load Balancer** Load balancer fordeler inngående trafikk så ikke enkeltressurser overbelastes. Den håndterer også utgående koblinger fra nettverket.

	**Network Security Groups** Network Security Groups kontrollerer trafikk til og fra ressurser i et virtuelt nettverk. Dette tillater ressurser på samme konto, i samme ressursgruppe og samme virtuelle nettverk å ha separat tilgangskontroll. Og gir en god sikkerhet.
	**Azure Active Directory** Azure Active Directory (AAD) er en skybasert innpakking av Active Directory (AD). Her håndteres brukere, grupper, autentisering (AuthN) og autorisasjon (AuthZ). AAD i denne «basisversjonen» er ment for å fungere som katalogtjeneste for én bedrift, og er ikke beregnet for eksterne brukere uten tilleggsfunksjonalitet. Tilleggsfunksjonalitet her vil være tjenester som Azure AD B2C (business to consumer) og Azure AD B2B (business to business).
	Transparent Encryption Alle data lagret i Azure SQL databaser er kryptert på disk, dette inkluderer loggfiler og foregår på sidenivå av dataene. Data krypteres før de skrives til disk, og dekrypteres når data leses inn i minne igjen.
	Data Lake Encryptipon Data i datasjø krypteres ved lagring til disk som standard. Det brukes tre ulike krypteringsnøkler for å kryptere og dekryptere data.
	Azure Storage Service Encryption Dette er krypteringstjenesten for Azure Blob og fillager. Bruker 256-bit AES kryptering.
	**Hashicorp Vault** Hashicorp Vault tilbyr kryptering som en tjeneste med sentralisert nøkkelstyring for å forenkle kryptering av data i transit og i lager på tvers av sky og lokale datasentre. Hashicorp Vault lagrer hovednøkkel i Azure Key Vault – men det er mulig å lagre denne i en egen HSM modul for ekstra sikkerhet. Hashicorp Vault brukes for alle pseudo ID og håndtering av dynamiske nøkler. Vault er valgt da den tilbyr langt mer funksjonalitet enn Azure Key Vault, og i tillegg som en ekstern krypteringstjeneste gir datasjøen et ekstra sikkerhetsaspekt.
	**Azure Monitor** Monitor er en kombinasjon av Log Analytics og Application Insight. Her kan man overvåke de ulike tjenestene man har i Azure, gjøre analyse på loggene eller sette opp handlinger som respons på hendelser – for eksempel autoskalere tjenester som sliter med ytelsen, eller sende sms/e-post hvis uatorisert bruker prøver å lese data fra sensitive systemer.
	**Azure Security Center** Security Center er hjertet av sikkerhetsløsningen i Azure. Den kan også utvides til å inkludere lokale ressurser ved hjelp av egne overvåkingsagenter. Her vurderes alle tjenester om de er sikre nok i henhold til beste praksis. Systemene overvåkes for potensielle trusler og kommer både med anbefalinger til handlinger, og alarmer ved antatte trusler. Det er mulig å respondere til alarmer eller anbefalinger direkte fra sikkerhetssenteret uten å måtte manøvrere seg fram til riktig ressurs manuelt.
	**Azure Sentinel** Azure sin Security Information and Event Managment (SIEM) tjeneste, legger kunstig intelligens og avansert funksjonalitet på toppen av sikkerhetsløsningene i Azure. En avansert tjeneste som styrker sikkerheten utover hva som er tilgjengelig i Azure Security Center og Azure Monitor. Dette er en anbefalt tjeneste for å sikre datasjøen.
	**Web Application Firewall** Web Application Firewall er en tjeneste som står mellom applikasjonslaget og betjeningslaget. Den analyserer henvendelser for å sikre at ikke ondsinnet kode blir sendt inn. Dette med tanke på å sikre mot angrep av type SQL Injection, Cross Site Scripting eller Denial of Service.
	**Azure IAM** Identity and Access management (IAM) i Azure beskytter virksomhetens programmer og data ved inngangsporten. IAM sørger for at virksomheten kan velge hvilke personer og maskiner som skal ha tilgang til de ulike programmene og dataene, samt hvilken type tilgang. IAM forsvarer virksomheten mot ondsinnede påloggingsforsøk og sikrer autentiseringslegitimasjon gjennom

	risikobaserte tilgangskontroller, verktøy for identitetsbeskyttelse og sterke autentiseringsalternativer, uten å påvirke tilgjengeligheten.
	**Azure Key Vault** Det er meget viktig å kunne håndtere kryptografiske nøkler og passord på en sikker måte i skyløsninger. Key Vault håndterer klargjøring, konfigurering, oppdatering og vedlikehold av nøkler og sikre passord og minimerer sikkerhetsrisiko og kompleksiteten i en slik sikker løsning. Key Vault vil kun brukes for å håndtere de nøkler/hemmeligheter som maskinkontoene skal bruke, samt hovednøkkelen til Hashicorp Vault.

2.3 Roller

Rolle	Beskrivelse	Annet
Stavanger kommune		
Behandlingsansvarlig – fagsystemer	Direktør, leder for fagområde Virksomhetsansvarlig.	Disse er dataeier for dataene som legges i datasjøen. All bruk av data i datasjøen skal godkjennes av riktig behandlingsansvarlig.
Systemansvarlig - Datasjø	Oversikt over funksjonalitet og videre utvikling av løsningen(e) som dekkes av dette rammeverket.	Har overordnet ansvar for datasjøen og dette styringsdokumentet. Personen skal påse at alle generelle regler følges og at løsningen blir et godt verktøy for Stavanger kommune.
Global administrator - Azure	Den eller de som innehar denne rollen har ansvaret for alt som foregår i Azure. Det er global administrator som styrer alle tilganger og rettigheter på kommunens Azure-konto og sørger for kontroll, sikkerhet og kostnader på alle systemene i skyen.	Azure har flere nivå på administratorkonto. Det øverste er global administrator som har full tilgang til alle systemer, uansett tilgangskontroll.
Sikkerhetsadministrator – Azure	Ansvaret for å overvåke implementeringen av sikkerhetsstrukturen i Azure. Ansvaret for å sette opp reglene for sikkerhetssenteret og oppfølging av disse. Ansvaret for krypteringstjenester.	Global administrator kan delegere ansvar for oppfølging og vedlikehold av sikkerhetsstrukturen til en dedikert person. Men typisk nok vil sikkerhetsadministrator og global administrator dele på arbeidet.
Sikkerhetsansvarlig	Sikkerhetsansvarlig har ansvaret for sikkerheten i organisasjonen, herunder IT-sikkerhet.	Ansvarlig for å gjennomføre eller påse at det gjennomføres ROS-analyser og DPIA ved behov.
Bestiller	Bestiller i dette tilfellet er den som ønsker å bruke data lagret i datasjø.	Vil være ansvarlige for å forklare behov for data og hvordan data vil bli brukt.
Dataforsker	Utforsker og analyserer data i datasjøen og skaper ny innsikt basert på dette.	Tilgang til deler eller alle data i datasjø.
Forretningsanalytiker	Kjenner ofte et spesifikt domene eller del av organisasjonen. Er ofte en	Tilgang til deler eller alle data i betjeningslaget.

	brobygger mellom forretning og teknologiløsninger og ønsker å effektivisere forretningen ved bruk av teknologi/data. Jobber ofte sammen med dataforsker og dataingeniør for å gi innsikt i forretningsbehov og for å hente ny innsikt fra data til nytte for forretningen.	
Personvernombud (DPO)	Personvernombud bistår med veiledning, rådgivning, kontroll og etterlevelse av personvernlovverket i Stavanger Kommune, i henhold til egen rollebeskrivelse jf Stavanger Kommunes organisajonskart og rolle- og ansvarsbeskrivelser.	Ingen tilgang til systemet, men kan ha behov for innsyn ved avvik eller utøvelse av registrertes rettigheter. Rådføres ved personvern spørsmål. Skal vurdere og gi sin kommentar til alle DPIA er som utføres.
Utviklere		
Kildesystem analyse	Analyse av kildesystemer og data i kildesystemene.	Tilgang til nødvendig kildesystem som skal analyseres.
Dataingeniør - mottak	Utvikler av mottak og lagring av data i datasjøen. Bruker Data Factory, Azure Functions, IoT Hub og Data Lake.	Tilgang til datasjø og Indirekte tilgang til kilder via Data Factory.
Dataingeniør - transformasjon	Utvikler Transformasjonsjobber - med særlig vekt på T for transformasjon - fra datasjø til betjeningslaget. Bruker Data Factory, Databricks og Stream Analytics.	Tilgang til å lage Databricks jobber og indirekte tilgang til datasjø.
Dataingeniør - betjening	Avhengig av hva som skal utvikles i betjeningslaget så vil denne rollen ha tilgang til nødvendige data i dette laget.	Tilgang til data i betjeningslaget.
Dataforsker	Bruker Azure Databricks til utforsking og analyse og ofte Azure Machine Learning Studio til avansert analyse.	Tilgang til deler eller alle data i datasjø.
Presentasjonsutvikler	Bygger løsninger for å presentere data fra betjeningslaget. Eksempel på løsninger er Front-end mobil og web, og Power BI rapporter.	Tilgang til relevant data i betjeningslaget.
Mottakere		
Mottakere	Alle brukere/systemer som får tilgang til data direkte eller via datasjø. I de fleste tilfeller vil dette være via presentasjonslaget/betjeningslaget.	Legg merke til at mottakere kan være av ulike typer. Noen mottakere vil være en bruker/individ, mens andre kan være et IT system som mottar/leser data.
Interne mottakere	Autentiserte brukere/systemer i Stavanger kommune.	
Eksterne - autentisert	Autentiserte brukere/systemer utenfor Stavanger kommune.	
Eksterne - anonyme	Anonyme brukere/systemer utenfor Stavanger kommune.	Eksempler på dette er åpne data som er delt eksternt.

2.4 Innebygd personvern og innebygd sikkerhet

Innebygd personvern er en helt essensiell og lovpålagt del av alle bruk av informasjonsteknologi. I et prosjekt hvor flere datakilder som inneholder personopplysninger importeres til en datasjø, kan det forutses at det skal behandles store mengder data, på en systematisk måte, om et stort antall registrerte, og ofte for analyserelatert formål. En behandling i stor skala kan gjøre det mulig å oppnå et omfattende bilde om de registrerte. Derfor er det viktig at det tas hensyn til personvernet ved løsningens planlegging, implementering og forvaltning.

I tillegg til å kreve god forutgående oversikt og kontroll over egne behandlinger og dataflyt før migrering, bør kommunen ha:

- Oversikt over hvordan data som skal overføres til datasjøen vil støtte generelle strategiske mål eller beslutningsprosesser
- Oversikt over hvilke typer og kategorier av data og registrerte som skal omfattes
- Oversikt over gyldige behandlingsgrunnlag for behandlinger og viderebehandlinger
- Oversikt over hvem data skal deles med og det aktuelle behovet for tilgang og brukerrettigheter
- Oversikt over hvor dataene skal behandles fysisk, hvilke typer behandlinger som skal gjennomføres og hvem som skal bistå med behandling (og eventuelt brukerstøtte), osv.

En slik utredning og kartlegging er kommunen sitt ansvar. Det er også kommunen sitt ansvar å sørge for riktig og tilstrekkelig informasjon til de registrerte, slik at de er bevisst at deres data skal behandles i datasjøen etter de blir anonymisert.

Videre skal også sikkerhet ivaretas i løpet av hele livssyklusen til dataene. Personvern og sikkerhet (både cyber- og informasjonssikkerhet) er to sider av samme sak, når det gjelder behandling av personopplysninger.

Beskrevet arkitektur tar inn beste praksis og internasjonale standarder sammen med robust teknologi og industriekspertise til å bygge en løsning som støtter etterlevelse av personvernkrav og god sikkerhet.

Innebygd personvern blir forklart gjennom syv prinsipper, herunder:

1. Vær i forkant, forebygg fremfor å reparere

For å tilrettelegge for en personvernvennlig løsning er det viktig å vurdere risikoene for personvernet så tidlig som mulig i utviklingsprosessen. Kostnadene ved å rette feil og mangler ved et ferdig system kan være høye. Ved å ta hensyn til personvernet tidlig i utviklingen kan unødvendige krenkelser av personvernet unngås.

Derfor er det viktig at Stavanger kommune utreder risikoene for sikkerhet og personvern knyttet til løsningen fra begynnelse og kobler dette arbeidet til planlegging og iverksettelse av datasjøen. Eksempler på risikoer som skal tas spesielt hensyn til er eventuell skjev maktbalanse mellom kommunen og kategorier registrerte, risikoer knyttet datadeling, risikoer knyttet til sammenheng, samt risikoer knyttet til omfang av behandlingen, slik som: Hvilke forventninger har de registrerte? Skal alle ansatte opplysninger gå inn i datasjøen? Er tenkt anonymisering sterk nok til å unngå re-identifikasjon av data hvis data blir sammenstilt med andre datakilder av tredjeparter? Hva er konsekvenser hvis det skjer re-identifikasjon, osv.

Bruk av samtykke som behandlingsgrunnlag for ansatte er kun lovlig i noen få tilfeller og må evalueres sammen med personvernombudet. Bruk av samtykke som behandlingsgrunnlag for innbyggere må vurderes nøye opp mot skjev maktbalanse og bør diskuteres med personvernombud.

2. Gjør personvern til standardinnstilling

Standardinnstillinger er førende for hvordan et system blir brukt og hvordan personopplysninger blir lagret. For at et system skal ha innebygd personvern må standardinnstillingene settes opp slik at de gjenspeiler den minste krenkende innstilling til de registrertes rettigheter og friheter (dataminimering). Et slikt system vil automatisk styre brukeren til en arbeidsmåte som gir bedre personvern.

I tillegg til etterlevelse av dataminimering, innebærer det f.eks. at metadata støtter utøvelse av registrertes rettigheter, at data slettes automatisk i datakilder og i datasjøen, at tjenstlig behov vurderes også på brukerrettigheters nivå og etter det som oppgaven krever (spesifikk tilgang), eller at det brukes personvernfremmende teknologi (PET), som f.eks. diverse anonymiseringsteknikker.

3. Bygg personvern inn i designet

Personvern skal være innebygd i IT-systemets design og arkitektur, samt forretningspraksis, istedenfor at funksjonene som sikrer overholdelse med personvernregelverket legges til i etterkant. Personvernet blir dermed en viktig del av kjernefunksjonaliteten.

4. Skap full funksjonalitet

Gjennom innebygd personvern ivaretar Kommunen både brukeren og registrertes personvern og egne behov. Det er viktig å ta hensyn til personvernet fra start for å unngå endringer som kan gå utover funksjonaliteten til løsningen. Noen ganger er det ikke mulig å gjøre endringer i etterkant uten at systemet blir dårligere. Målet er en både og tilnærming fremfor en enten eller, for å ivareta virksomhetens behov og interesser og brukernes og registrertes personvern.

5. Ivareta informasjonssikkerheten fra start til slutt

Informasjonssikkerhet må være en del av løsningen. Det betyr at alt som skjer i systemet på forhånd er risikovurdert og hensiktsmessig sikret, helt fra før personopplysningene samles inn, mens de behandles, og til de er slettet. Personopplysningene skal sikres mot uautorisert tilgang, endring, ødeleggelse og spredning.

I løsningen, tas det en helhetlig tilnærming til sikkerhet omtales som samtidig er spesifikk til de behovene til hver komponent av datasjøen.

6. Vis åpenhet

Brukerne og registrerte skal både kunne oppfatte systemet som transparens. Det skal være mulig å kontrollere at lovverket og interne føringer etterleves, og hvordan personvernet blir ivaretatt. Kommunen skal sørge for at brukerne får god informasjon og at det legges til rette for innsyn i egne opplysninger. Samtidig skal det være mulig å kontrollere at systemet ivaretar personvernet slik leverandøren oppgir.

7. Respekter brukerens personvern

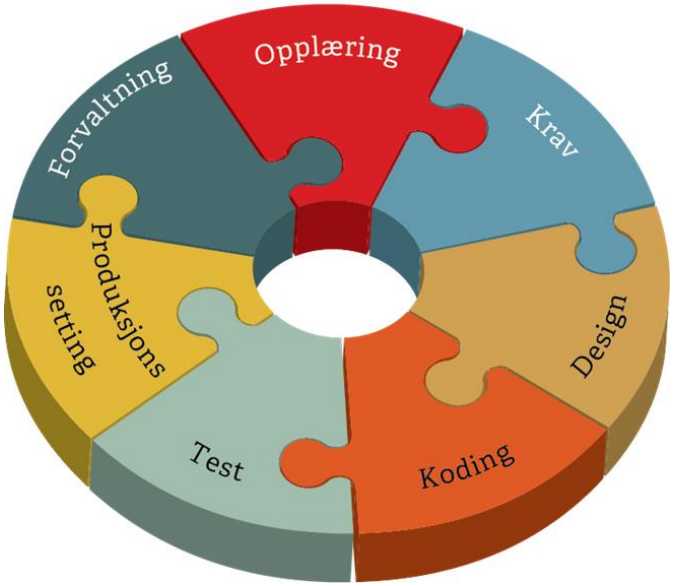
Fremfor alt krever innebygd personvern at utviklerne, bestillerne og administratorer gir de registrertes personvern høy prioritet. Dette vil si å sørge for at de registrerte kan utøve sine rettigheter og får tilstrekkelig informasjon om behandling for å kunne ivareta egne rettigheter. Det er viktig at alle faser som design, konfigurasjon, testing med mer går gjennom konsekvensutredninger som dekker kvalitet, sikkerhet og forretningsmessige aspekter og behov i tillegg til behandling av personopplysninger.

Disse prinsippene krever i praksis at personvern og sikkerhet ivaretas i Kommunes egne arbeidsmetodikk og rutiner, samt i løsningen selv. Uten god opplæring innen disse områdene og en forankring av personvernet og sikkerhetsarbeidet i ledelsen kan brukerne utsette løsningen for utilsiktede sårbarheter.

Disse prinsippene var også grunnmur til Datatilsynets veiledning om programvareutvikling med innebygd personvern, som utvider på hvordan virksomheter kan sørge for en personvernvennlig tilnærming til utvikling, implementering og forvaltning av IT-løsninger, som datasjøen. På andre ord, veiledningen gir føringer og sjekklister for operasjonalisering av innebygd personvern og innebygd sikkerhet.

2.4.1 Personvern innebygd i løsningen som leveres

Innebygd personvern og personvern som standardinnstilling krever at alle adekvate risikoreduserende tiltak er tatt i betraktning i alle utviklingsfaser av datasjøen. Datatilsynets retningslinjer og sjekklister for innbygget personvern bør introduseres gjennom alle faser av prosjektet. Dette inkluderer konfidensialitet, integritet, tilgjengelighet og robusthet, autentisering, autorisasjon, logging, kryptering med mer.



'Det er opp til den enkelte virksomhet å vurdere hvilken metodikk som skal brukes, hvilke tiltak, områder og faser som skal vektlegges og hvor innsatsen skal økes. Valg av metodikk vil typisk påvirkes av tjenesteområde, virksomhet, type programvare som skal utvikles, og betraktninger rundt og oppfattelse av egen risiko'

Datatilsynet

2.4.1 Operasjonalisering av innebygd personvern og innebygd sikkerhet

Som forklart ovenpå skal man utrede risikoer knyttet til implementering av datasjøen og holde dette risikovurdering oppdatert. Som en del av dette arbeidet skal det tas hensyn til oppfyllelse av personvernkrav og andre relevante lovkrav (*compliance* risikoer) og verifisere for eksempel om det er plikt til å gjennomføre en personvernkonsekvensvurdering (DPIA). Ved utarbeidelse av DPIA skal kommunens personvernombud involveres. Som en ledd i denne vurdering, skal Stavanger kommune utarbeide en tydelig skisse av dataflyt som forventes etablert og relatert tjenstlig behov. Like viktig er da å utrede potensielle mottakere til opplysninger og integrasjoner. Det bidrar til å gi et mer helhetlig risikobilde og sørge for bedre oversikt.

Viktige elementer å ta hensyn til omfatter også ukjent datakilder, data som er behandlet utenfor formelle rutiner, en utredning om det foreligger mer data enn det er behov for i systemene eller om brukernes forretningsbehov ikke er dekket fullt ved tilgjengelige data, manglende data kvalitet, osv.

Risikoarbeidet bør i etterkant gjennomgås med valgt leverandør for å sørge at planlagt tiltak er tilstrekkelig for å dekke behovet.

Datatilsynets veiledning og sjekklister for innebygd personvern bygger et godt rammeverk for å dokumentere arbeid som gjennomføres. Stavanger kommune anbefales også å vurdere behov for videre opplæring av ansatte knyttet til prosjektet. Med god intern kompetanse kan kommunen bygge et godt grunnlag for både opprettelse og forvaltning av løsningen.

Kommunen skal sørge for egen overholdelse av personvern lovkrav og en forsvarlig informasjonssikkerhet og cybersikkerhet praksis. Herunder, men ikke uttømmende, skal overføringer krypteres og gjennomføres gjennom sikre kanaler, robusthet skal testes tilstrekkelig, anonymisering testes og vurderes hensiktsmessig utført, og tilgangsstyring og -kontroll opprettes etter godkjent tjenstlig behov. Videre skal det testes funksjonelle og ikke-funksjonelle krav som dekker sikkerhets- og personvernbehov – f.eks. om tekniske tiltak for å sikre håndheving av de registrertes rettigheter er effektive. Disse elementene diskuteres i mer detaljer gjennom dette dokumentet, der hvor diskusjonen er relevant, men følgende gir en overordnede bildet.

Datasjøen skal ivareta innebygget personvern i fire områder: miljø, bruk, etterlevelse (compliance) og bærekraft. Disse gjør det mulig å skape «Trust by Design» (innebygd personvern, sikkerhet og etikk):

Miljø
I miljøområdet skal kommunen ta spesielt hensyn til hvordan løsningen samhandler med og kan påvirkes av tilkoblingen til Internett, andre løsninger, utstyr, brukere og så videre. Samt at tekniske og organisatoriske tiltak omfatter hele livssyklusen. Balansen mellom tilgjengelighet og konfidensialitet er her spesielt viktig.

Bruk
I brukerområdet fokuseres det på potensielle skader på helsen, sikkerheten og personvernet til den registrerte og brukeren, og dens eiendom; det er skapt brukervennlig og universell utformet design med personvern som innstilling, og det er gitt full funksjonalitet. Krypterte kommunikasjonskanaler, tilgangsstyring og bruk av anonymiseringsteknikkene har en sentral rolle i reduksjon av risikoen for berørte individer.

Etterlevelse
I etterlevelseområdet (compliance) sørges det for at løsningen etterlever lovkrav. Løsningen skal også sørge for at Kommune kan svare på registrertes forespørsler innen lovens definerte frister. Innhentning av data skal også alltid formålsprøves. Beste praksis er tatt i betraktning.

Bærekraft
I bærekraft tas det hensyn til hvordan løsningen kan bidra til å forsterke tilliten brukeren og den registrerte har i Kommunen. Løsningen har forebyggende håndtering av sensitive eller særlige kategorier av personopplysninger ved å sikre at disse kategorier personopplysninger identifiseres og omhandles med ekstra vern. Løsningen ivaretar også universell utforming og sørger for at den kan håndteres av ikke-tekniske brukere på alle ender av løsningen.

Del 1 - Hente data fra kilde og lagre i datasjø

3 Kildesystemer

Stavanger kommune forholder seg til en stor mengde kildesystemer. Kildesystemer defineres som alle systemer der Stavanger kommune er dataeier. Selv om ikke alle kildesystemer forvaltes av Stavanger kommune, er de allikevel definert som kildesystemer og er derfor videre kandidater for integrasjon til datasjø.

Noen kildesystemer er forvaltet av Stavanger kommune, men selve dataene eies av Rennesøy- eller Finnøy kommune. Dette dokumentet definerer kildesystemer som alle systemer som Stavanger kommune har et forhold til, uavhengig hvor det originalt hører til. Dette i tråd med 'Nye Stavanger' og kommunesammenslåingen i 2020.

Hvert kildesystem vil måtte vurderes individuelt med tanke på strategi for håndtering i forhold til datasjø og videre bruk til analyse. Unntak fra reglene beskrevet under vil eksistere. Det er svært viktig at kildeanalysen avdekker disse, og dokumenterer hvorfor man har unntak.

Dette rammeverket dekker ikke forvaltning- og/eller sikkerhetsregler som allerede eksisterer på de ulike kildesystemene. Men dette er regler som analyse av kildesystem vil måtte ta hensyn til. Det å kopiere data til datasjø, kan medføre endringer i forvaltning- og/eller sikkerhetsregler i kildesystemene.

Kildesystemer og data i kildesystemer faller inn under ulike kategorier når det kommer til sikkerhet. Det er viktig at alle sikkerhetsavklaringer er gjort tidlig for å sikre at videre tiltak blir riktige.

3.1 Prinsipper

- Analyse av kildesystem skal være så detaljert at man kan implementere løsninger som kopierer data til datasjø, og håndtere data i henhold til sikkerhetsdefinisjonen til kilden.
- Analyse av - og tilgang til - kildesystem skal være godkjent av behandlingsansvarlig.

3.2 Kategorisering

Av de kjente kildesystemene vil dette dokumentet gjøre en grov inndeling og kategorisering av disse systemene for å gi en veiledning på hvordan de bør behandles som kilder. Den individuelle analyse vil gi de konkrete behovene for hvert system. Kildesystemer kan falle under en eller flere av disse og det vil alltid bli en individuell vurdering på hvordan data skal håndteres.

3.2.1 Gruppe 1 – Historisk/Aktive Data

Gruppe	Beskrivelse	Standard Behandling/Bruk
Historiske Data	Dette er systemer som ikke lenger er i bruk eller som innen kort tid vil bli avsluttet eller tatt ut av bruk.	Data flyttes en gang til sky og gammelt kildesystem kan avvikles.
Aktive Data	Dette er systemer som brukes aktivt av tjenesteområdene.	Data kopieres til sky med en definert frekvens. Alle data er i utgangspunktet ikke nødvendige, men så mye som mulig skal kopieres over og dokumenteres.

3.2.2 Gruppe 2 – System lokasjon

Gruppe	Beskrivelse	Standard Behandling/Bruk
On-Premises	Systemer som forvaltes av Stavanger kommune eller som Stavanger kommune enkelt kan koble seg på for uttrekk av data.	Sette opp Data Factory jobber som henter data på en gitt frekvens fra kildesystemet.
ASP	Systemer som forvaltes av tredjeparts leverandører.	Avhengig av hva leverandør av løsning eller forvalter tilbyr, vil man ha ulike muligheter for integrasjon En-gangs dump av DB (bare aktuelt for historiske system) Direkte oppkobling til DB for uttrekk API (Anbefalt) Andre varianter for dumping av data (FTP, Excel, Mail osv.) RPA for å lese data ut av system.
IoT	Systemer som håndterer IOT.	Event Hub for mottak av data.
Stormaskin	Tidlige versjoner av transaksjonssystemer der bruker er separat fra selve maskinen. Bruker aksesserer maskinen via egne terminaler eller hullkortlesere.	Lese data fra database på stormaskin via egnede verktøy.
Annet	Data som faller utenom de overnevnte kategorier. Dette kan f.eks. være filer i en fil-struktur.	Individuell analyse og håndtering må gjøres basert på det aktuelle behovet.

3.2.3 Gruppe 3 – Data typer

Gruppe	Beskrivelse	Standard Behandling/Bruk
Strukturerte data	Data som har en definert datamodell og som er lesbart av maskin. Det vanligste er data som ligger i databaser, i tillegg inkluderer dette filer som XML, CSV, JSON.	Hovedregelen er å lagre dette i datasjø. For noen unntak kan det argumenteres for å lagre dette på en relasjonsdatabase i datasjøen.
Ikke strukturerte data	Data som verken har en predefinert datamodell eller er organisert etter en forhåndsdefinert struktur. Typiske eksempler her er Word, PDF og bilder.	Hovedregelen er å lagre dette i Blob Store.
Sanntidsdata	Data som for eksempel kommer i en strøm fra gjenstander tilknyttet internett.	Event Hubs tar imot datastrømmen.

3.2.4 Gruppe 4 – Dataklassifisering

Gruppe	Beskrivelse	Standard Behandling / Bruk
Personopplysninger	Alle opplysninger og vurderinger som kan knyttes til enkeltpersoner direkte eller indirekte. De aller fleste kildesystemer vil inneholde en eller annen form for bruker/personopplysninger.	Disse dataene skal alltid behandles i samråd med prinsippene i GDPR. Se kapittel *18.1*. Hovedregel er at disse data alltid skal anonymiseres når de blir brukt til analyseformål.
Særlige kategorier personopplysninger	Opplysninger som krever ekstra vern, slik som rasemessig eller etnisk opprinnelse, religion, helseopplysninger og seksuell orientering mm. Disse er fastsatt i GDPR artikkel 4. Tilgjengelig kun for autentisert og autoriserte brukere i Stavanger kommune. Dette er kun brukere som har tjenstlige behov som skal benytte denne typen data, og all bruk skal være godkjent av dataeier.	Som hovedregel er det forbudt å behandle disse dataene. Regelen har imidlertid flere unntak. Se Datatilsynet. Ved evt. unntak er hovedregel at disse data alltid skal anonymiseres når de blir brukt til analyse eller annet formål. I tillegg er det andre personopplysninger som av mange anses som sensitive/følsomme, eks.

		lønnstrekk, kredittvurdering eller personnummer. Hovedregel om forbudt behandling gjelder ikke her, men det anbefales at disse følger beskyttelsesregimet til særlige kategorier og anonymiseres alltid ved høsting av innsikt. Begrepet sensitive personopplysninger brukes derfor i dokumentet for å omfatte både disse to kategorier.
Unntatt offentlighet	Saksforberedelser, bedriftskonfidensiell informasjon, informasjon som berører samfunnssikkerhet etc.	Hovedregel er at disse data alltid skal anonymiseres når det blir brukt til analyse. Ved bruk av data som er unntatt offentligheten skal det gjøres en særlig vurdering om det er nødvendig, lovlig og forsvarlig å bruke disse dataene selv om de anonymiseres. Bruk og videre offentliggjøring av data eller resultatet av data som er unntatt offentligheten kan ha store skadevirkninger for individer og samfunnet selv om de er anonymisert.
Internt	Åpent for alle autentiserte brukere i Stavanger kommune.	
Åpen	Åpen for alle (i hele verden).	

3.3 Mål og leveranser i analysearbeid

Analyse av kildesystemer skal i all hovedsak løse to oppgaver

1. Gjøre data tilgjengelig i datasjø
    - Hente ut data fra kildesystem og lagre i datasjø
    - Bestemme hvordan det skal lagres i datasjø
    - Definere hvor ofte data skal hentes fra kilden
2. Dokumentere data i datasjø for å tilrettelegge videre arbeid
    - Dokumentere data fra kildesystemet i Datakatalog
    - Analysen vil føre til forståelse og oppdagelser i kildesystemet som kan være relevant for senere bruk av data fra datasjø. Dette skal dokumenteres.

3.3.1 Leveranser



3.3.1.1 Leveranse 1 – Forståelse og omfang av kildesystemet

Dette blir den første fasen der målet er å få en overordnet forståelse av kilden. Dette vil hjelpe å gi føringer for de andre leveransene.

Hva	Beskrivelse	Hvorfor
Forretningsområde / Tjenesteområde	Tjenesteområdet kilden brukes til.	Må finne ut hvor kilden hører til i Stavanger kommune, og hva den brukes til for å kunne sette data i kontekst.
Forretningsprosess	Overordnet forretningsprosess, og hvilke deler av som dekkes av kilden.	Etablere forståelse for hvor systemet hører inne blant andre systemer.
Interessentanalyse	Hvem er behandlingsansvarlig? Hvem er systemansvarlig? Hvem er mottakerne? Har vi andre sentrale/relevante kontaktpersoner? Hvor ofte blir systemet brukt? Når er belastningen størst?	Etablere oversikt over hvilke personer som kjenner kildesystemet, og hvem som kan gi tilgang til systemet, forstå data og hva data skal brukes til videre.
Informasjons elementer	Datagrupper/datatyper som ligger i kilden, og hva er de viktigste dataene og attributtene. Andre datakarakteristikker: Datakvalitet Datoformater Konverteringsregler	Etablere forståelse av hva som ligger lagret og få oversikt over de mest sentrale informasjonselementene. Utpeke hvilke data som er relevante for tjenesteområdet og avgrense hvilke data som skal lastes til datasjø. Dette er viktig for å spisse dataanalysen.
Sikkerhet	Sikkerhetsaspekter knyttet til kilden og hvordan tilganger blir håndtert og tildelt.	Viktig informasjon for å koble til kilden og kunne kopiere data til datasjø, samt valg av Data Factory eller Data Migration Assistant for kopiering til datasjø.
Driftsmiljø	Informasjon knyttet til drift av løsningen. Eksempel på relevant informasjon er: Hvordan løsningen forvaltes Hvilken teknologi som brukes Hvilke deler av systemet som er i bruk Integrasjoner til andre systemer Eksisterende rapporter i løsningen eller basert på data fra løsningen Brukerdokumentasjon og annen relevant dokumentasjon	Dette gir innsikt i hvordan data best kan lastes til datasjø, hvilken dokumentasjon som finnes og vil være nyttig for dataanalysen og hvilke data som skal hentes basert på hvilke deler av systemet som er i bruk.
Andre begrensinger og antagelser	For eksempel kjente feil og mangler med systemet, og datakvalitetsproblemer.	Avdekke kjente feil/mangler som man må fikse før flytting, eller som vil skape problemer senere i prosessen. Dersom feil og mangler er begrenset til deler av systemet, kan det påvirke hvordan dataanalysen skal struktureres.
Framtidige endringer/behov	Planlagte og/eller forventede endringer i kildesystemet	Skape oversikt over kommende endringer som er relevante for hvordan integrasjon til kilden settes opp. Dette kan brukes som input til *3.3.1.4 Leveranse 4 – Kilde dokumentasjon*

3.3.1.2 Leveranse 2 – Dataanalyse

Teknisk analyse av systemet for å analyserer hele eller deler av kilden.

Før man begynner må personen som gjennomfører analysen ha tilganger til kilden. For at dette skal være gjort på best mulig måte, må det være akseptert at denne personen kan få innsyn i sensitive data. Oppgaven kan utføres av en person som allerede har riktige tilganger, hvis ikke må personen godkjennes av systemansvarlig/behandlingsansvarlig.

Hva	Beskrivelse	Hvorfor
Datamodellerings-prinsipper	Hvordan data ligger lagret i systemet, f. eks. på en database, på et filsystem eller andre måter. I tillegg, og hvilke designprinsipper som er brukt ved lagring, f. eks. normaliserte data og fremmednøkler.	Forstå hvordan datakilden har strukturert dataene for å finne ut hvordan de henger sammen.
Gjennomgang av data	Oversikt over alle tabeller, filer osv. som finnes i kilden.	
Data i tabeller	Analysere innholdet i tabeller og attributter Hva er masterdatatabeller? Hva er transaksjonstabeller? Hva er oppslag eller konverteringstabeller?	Relevant for dokumentasjon i data katalog. Videre er det relevant for bruk av data til analyse der data potensielt må transformeres til felles format.
Data formater	Forstå hvordan data er lagret Dato formater / Kalender definisjoner Konverteringsregler (valuta, enheter) Nummerformat (skilletegn) Nåværende definert formatter og konsistens av denne formatteringen på tvers av datakilder.	Mest relevant for bruk av data til analyse der data potensielt må transformeres til felles format.
Volum	Opprette forståelse av volum og størrelse av data ved å vurdere: • Antall registreringer • Antall filer • Antall tabeller • Størrelse på filer/tabeller	Kan brukes til å vurdere om alle data skal flyttes.
Datakvalitet (og data integritet)	Forsøke å identifisere svakheter i dataene. Fines det eksisterende kontroll/sjekk på datakvalitet?	Mest relevant for bruk av data til analyse der data potensielt må vaskes og sjekkes. Dårlig datakvalitet i kilden vil føre til dårlig datakvalitet i analysen og gir da dårlig beslutningsgrunnlag.
Inn / Ut	Finnes det eksisterende integrasjoner med andre systemer? Gjøre det uttrekk til rapporter både rett i løsninger, men også i felles rapporteringsverktøy?	Det finnes kanskje andre systemer man er avhengig av for å forstå alle dataene. Det foreligger eksisterende datatransformasjoner som kan være relevante. Det ligger logger/filer utenfor kilden som man ønsker å ha med i datasjø.
Logger	Hvor ligger logger lagret? Ligger det relevant informasjon i loggene?	Er det relevant å ha med logger til datasjø.

3.3.1.3 Leveranse 3 – Migrering/Integrasjon design

Basert på tidligere leveranse kan man nå ta beslutninger på hvilke data i kilden som skal kopieres til datasjø og hvordan dette skal gjøres.

Design skal beskrive følgende:

Hva	Beskrivelse	Annet
Tilgjengelighet til kilden	Hvordan skal data aksesseres fra Data Factory?	Ikke alle kilder vil ha mulighet for en direkte kobling til kilden. Kilder kan ligge eksternt eller i et slik format at direkte kobling ikke er mulig. Da vil man måtte gjøre noe på selve kilden som gjør data tilgjengelig.
Data uttrekk fra kilden	Hvilke data skal hentes? Trenger vi alt? Husk spesielt: • Logger/Journaler • Ustrukturert data i database eller filstruktur	Hovedprinsipp er at alt skal kopieres til datasjø. «Alle data» er for udefinert og det må designes i detalj hvilke data det er snakk om. Det er veldig viktig at man beskriver hvorfor data er viktig i datasjø og hva formålet er.
Transformering	Må vi bearbeide/endre data før det lagres i datasjø?	Hovedregel er at man IKKE skal endre data, og at data lagret i datasjø skal være likt eller så nært kildesystemet som mulig.
Lagring	Hvordan skal data lagres i datasjø	Se senere seksjoner for regler.
Sikkerhet	Hvordan skal data sikres i datasjø?	Se senere seksjoner for regler.

3.3.1.4 Leveranse 4 – Kilde dokumentasjon

Når man har bestemt hvilke data som skal over i datasjø, vil man vite hva som er nødvendig å dokumentere. De andre leveransene er dokumentasjon i seg selv, men det som dekkes i denne leveransen er i all hovedsak det som skal dokumenteres i Datakatalog. Målet med denne leveransen er å jobbe strukturert med analyse av kilden, slik at man er best mulig forberedt på det som skal inn i Datakatalog.

For detaljerte regler for Datakatalog, se *0)*, så må det være minst fem personer (K-anonymitet) som har det samme kjønnet, bilmerke og postnummer, for at det skal være anonymt.

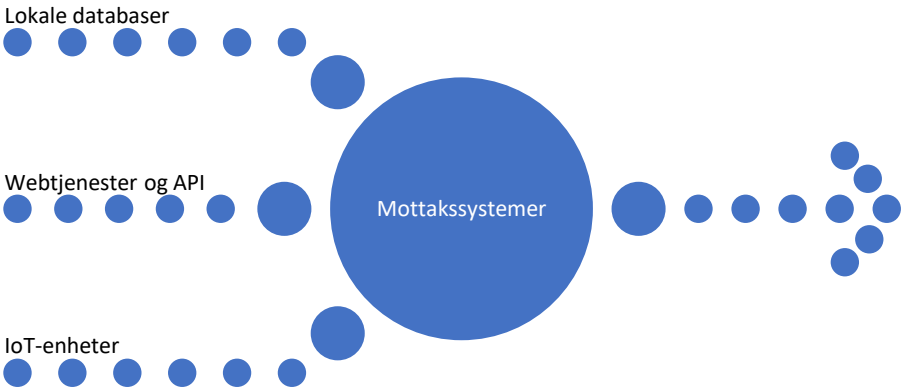
Nivået som kreves for å anonymisere avdekkes i analysefasen og gjennom bruk av ARX risikovurdering, dette må vurderes i hvert tilfelle.

En god guide til anonymisering fra det engelske datatilsynet: https://ico.org.uk/media/1061/anonymisation-code.pdf

Datakatalog.

4 Mottak

Med mottak menes de tjenester som fungerer som inntak til datasjøen. Dette kan være tjenester som passivt mottar data fra kildesystemer og sender de videre, for eksempel Azure Event Hub, eller det kan være tjenester som aktivt henter data fra kildesystemer og så sender de videre til lagring.


Lokale databaser
Webtjenester og API
IoT-enheter
Mottakssystemer

4.1 Systemer

For mottak av data velges det mellom 3 ulike systemer:

Azure Data Factory (ADF): I tillegg til at Data Factory er et orkestreringsverktøy som kontrollerer kall av ulike system/tjenester for flytting og transformasjon av data fra kilde til betjeningslaget, har Data Factory også funksjoner for å flytte data fra diverse kildesystemer (de fleste relasjonsdatabaser, flatfiler ++) til skyen.
ADF er det foretrukne mottakssystemet i Stavanger kommunes datasjø. Visse unntak finnes, som når man skal foreta engangs-kopiering av kilder (flytting av historiske kilder), når data kommer inn i sanntid (events) eller i de tilfellene der ADF ikke har de nødvendige koblingene.

Siden de ulike jobbene i Data Factory har sine egne koblinger og pipelines, blir det fort veldig uoversiktlig med for mange pipelines. Derfor skal som et minimum hvert enkelt fagområde ha sin egen Data Factory instans. Viktige kilder bør tildeles egne instanser. Den enkelte pipeline skal legges i mappestruktur basert på kildesystem.

Azure Event Hubs er laget for å motta data som kommer inn som hendelser i nær sanntid.
Azure Event Hubs deles i to systemer – Azure Event Hub og Azure IoT hub.

For alle kilder som kun skal sende data, og der det ikke er behov for verifisering av sensorer/enheter, skal **Event hub** benyttes, da denne er rimeligere enn IoT Hub.

Azure IoT hub benyttes der det er behov for økt sikkerhet – autentisering per enhet – og der det er behov for toveis kommunikasjon med enheten. Typisk vil dette være et behov der man mottar data fra spredte sensorer som kommunen selv trenger å vedlikeholde.

Azure Functions er serverløse applikasjoner som kan utføre en rekke oppgaver. I mottakslaget vil Azure Functions være et alternativ der hverken Azure Data Factory eller Azure Event Hubs har løsninger som passer kildene. Et eksempel her kan være en ASP-leverandør med Azure AD B2B innlogging til API.

Azure Logic Apps er som kan brukes sammen med, eller i stedet for Azure Data Factory. Logic Apps har bedre støtte for API-kall enn Data Factory. Både Logic Apps og Data Factory kan kalle hverandres tjenester – og det vil være behov for begge tjenestene.

Data Migration Assistant er et program som kjører på en lokal maskin og benyttes i de tilfellene man skal gjøre en engangskopiering av en relasjonsdatabase til Azure. Den skal kun benyttes for kopiering av historiske systemer der det ikke vil være behov for å kjøre importen på nytt.

4.2 Prinsipper

- Data i kildesystem må kunne aksesseres av Data Factory
- Passord og påloggingsinformasjon skal aldri legges direkte inn i Data Factory, de skal alltid legges i Azure Key Vault og så hentes derfra.
- Data skal ikke hentes med en høyere frekvens enn data blir oppdatert i kilde eller bruk av data senere i prosessen
- Uttrekk skal ikke belaste kildesystemet unødvendig
- Alle data fra kilden skal være komplett
- Alle mottakssystemer skal logges og overvåkes for å oppdage feil
- Hvis lesing fra kilde eller skriving til datasjø feiler, skal hele prosessen stoppe
- Påloggingsinformasjon til aktuelle kildesystem skal legges i Azure Key Vault
- Data skal ikke transformeres ved flytting/kopiering til datasjøen
- Mottakstjenesten skal kun ha lesetilgang til kilde
- Mottakstjenesten skal kun ha skrivetilgang til lagring
- Hvis kildeanalyse tilsier at data er personsensitive, skal data krypteres fra de tas inn i mottakslaget i datasjøen

4.3 Sikkerhet

Tjenestene for mottak følger sikkerhetsoppsettet spesifisert under arkitektur, og ellers under prinsippene nevnt ovenfor.

5 Lagring

5.1 Systemer



Azure Storage (hierarchical namespace) er hovedtjenesten for datasjøen. Her lagres alle filer mottatt fra kildene med noen få unntak:

- Kilder som har blitt identifisert som nødvendig å lagre i relasjonsdatabase
- Data livstidsadministrasjon skal innføres, og det må være rutiner på plass for å analysere datalager. Og data som ikke har blitt benyttet (aksessert) de siste 5 år skal flyttes til kaldlager (Data Lake Storage Cold tier). I tillegg til at data som i kildeanalysen vurderes å kunne gå direkte til kaldlager skal sendes dit.

Azure Storage (BLOB store) er tjenesten som benyttes ved behov for å dele filer uten tilkobling til Active Directory. Dette er samme tjeneste som nevnt over, men med litt ulike APIer – litt tregere, litt billigere lagring, og med delt tilgangsnøkkel i stedet for Active Directory kobling.

Azure SQL DB er tjenesten som benyttes for å lagre data i relasjonsdatabase. Dette er en dyr tjeneste, og skal kun benyttes dersom kildeanalysen tilsier at det er nødvendig.

Systemene beskrevet over vil også kunne finnes igjen i betjenings-laget. Alle prinsipper og beskrivelser i dette kapittelet vil kun omhandle bruk av tjenestene for rådata. Prinsipper og beskrivelser av tjenestene i betjeningslaget vil finnes i kapittel *8, Betjening*.

5.2 Prinsipper

- Data skal lagres i forhåndsdefinert struktur basert på sikkerhetsklassifisering
- Alle data skal separeres fra hverandre
- Data fra ulike kildesystemer skal separeres fra hverandre
- Brukerkontoer skal hverken ha lese- eller skrivetilgang til datasjøen
- Filer lagret i datasjø skal være CSV eller Parquet
- Alle sensitive data skal lagres kryptert
- Ingen data klassifisert som sensitive kan lagres uten å være pseudonymisert, og kryptert på applikasjonsnivået i tillegg til datasjøstandarden med kryptert disk (Encryption at Rest)
- Alle personopplysninger som tas inn i datasjøen skal logges i brukerloggdatabasen med informasjon om hvilke data om hvilke personer som tas imot og hvor disse dataene lagres

5.3 Om datasjøen

Stavanger kommunes datasjø skal lagre kildedata i tilnærmet opprinnelig tilstand. I datasjøen vil hver kilde ha sin egen katalogstruktur. Der hvor kilden var en relasjonsdatabase skal hver tabell lagres som en egen CSV-fil. I enkelte tilfeller kan det av funksjonelle årsaker være nødvendig å lagre kilder i andre format, til og med i en relasjonsdatabase. Men da kun ved spesielt dokumenterte behov.

Et problem som rammer mange datasjø-prosjekter, er at mengden data som lagres etter hvert blir så stor at vi slutter å snakke om en datasjø og begynner å snakke om en datasump.

Stavanger kommunes datasjø vil derfor ha en datasjø som skiller mellom aktive, inaktive og unødvendige data. Data deles inn basert på livssyklusen dataene befinner seg i og typen data, se delkapitlene *5.5* og *5.6*.

5.4 Katalogstruktur

Katalogstrukturen i Data Lake Store – som er kjernekomponenten i kommunens datasjø – skal være delt opp som følger:

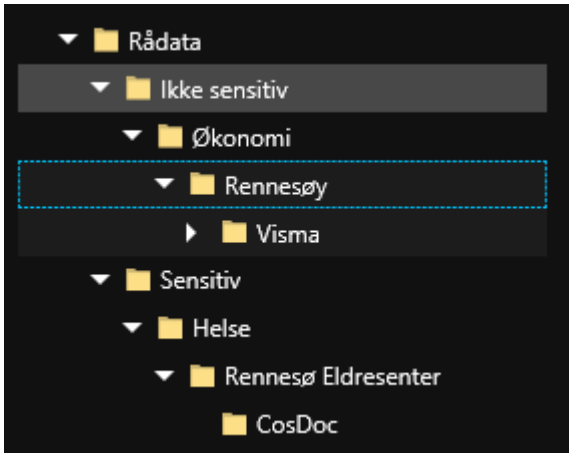
Hovedkatalog: Rådata

Underkataloger nivå 1: Sensitive og Ikke-sensitive

Underkataloger nivå 2: Fagområde/Forretningsområde

Underkataloger nivå 3: Institusjon/Avdeling

Underkataloger nivå 4: Kildesystem



Tilgang til de enkelte katalogene i datasjøen skal være underlagt streng kontroll, styrt av tilgangsgrupper som global administrator og sikkerhetsadministrator vedlikeholder.

I tillegg skal sensitive data være pseudonymisert og kryptert på applikasjonsnivå. Dette kommer i tillegg til Azure sin standard «encrypted at rest» funksjonalitet som vil si at disken dataene ligger på er kryptert.

Prinsippet om at kun maskinkonti har tilgang til rådata i datasjøen gjelder, men det må etableres kontroll for sjekk av hvem som kan få disse data levert til betjeningslaget uten anonymisering.

Unntaket her vil være i utviklingsmiljø, hvor det er nødvendig for dataingeniørene som jobber med mottak og transformasjon å utvikle løsningene. I utviklingsmiljø skal dataingeniører fortrinnsvis jobbe med testdata. Produksjonsdata skal ikke brukes i Utvikling og test uten at det er drøftet med kommunens personvernombud.

5.5 Aktive data

Data går igjennom en livssyklus. I denne konteksten er det antatt at livssyklusen til data ivaretas i kildesystemene og er i seg selv ikke dekket av dette dokumentet. Allikevel må dette ivaretas i overføring mellom de ulike kilde systemene og datasjø slik at vi kan sikre videre håndtering av data. En datasjø har som grunnprinsipp at alle data er av interesse i analyse- og innsiktsøyemed. Men ikke alle data er like interessante, og derfor defineres aktive data som data brukt i analyseøyemed eller i oppslag de siste 3 år (36 måneder).

5.6 Inaktive data

Data som ikke har vært i bruk de siste 36 måneder defineres som inaktive og bør flyttes fra varm-lagring (Hot tier storage) til kald-lagring (Cold tier storage). Både for å unngå datasump-problematikk, men også fordi det er billigere å lagre data i kald-lager.

Det er fortsatt mulig å gjøre oppslag og utforske inaktive data, inaktive data skal fortsatt ligge i datakatalogen. Men ettersom respons-tiden på kald-lager er høyere enn varm-lager, og pris per spørring er høyere så skal det også være en rutine på å fange opp data som blir brukt oftere enn 2 ganger over en 4 måneders periode og flytte disse tilbake til data lake store.

Tilsvarende er det naturlig å ha en øvre grense for hvor lenge data blir liggende inaktive før man tar en avgjørelse om de i det hele tatt skal tas vare på. Her bør man ha en manuell rutine hvor data som har vært

inaktive mer enn 5 år blir tatt opp til vurdering for sletting. Det bør vurderes opp mot eventuelle lovpålagt lagringskrav, og vurdering av innsiktsverdi av dataene.

Data som ikke merkes for sletting etter 5 år, og ikke har vært benyttet kan overføres til arkiv-lager (Archive tier storage). Dette er billigste lagringsmedium, men det koster ekstra å hente data ut fra arkiv-lager, i tillegg til at det tar tid – så data som blir spurt etter i arkiv-lager mer enn en gang bør vurderes flyttet til et annet lager.

Data klassifisert som sensitive må håndteres spesielt, da det fortsatt vil være spesielle sikkerhetsbehov for disse dataene selv om de er inaktive.

5.7 Dokumentasjon av data i datasjø

Data som legges i datasjøen skal dokumenters i datakatalogen. For mer informasjon se *0)*, så må det være minst fem personer (K-anonymitet) som har det samme kjønnet, bilmerke og postnummer, for at det skal være anonymt.

Nivået som kreves for å anonymisere avdekkes i analysefasen og gjennom bruk av ARX risikovurdering, dette må vurderes i hvert tilfelle.

En god guide til anonymisering fra det engelske datatilsynet: https://ico.org.uk/media/1061/anonymisation-code.pdf

Datakatalog.

Del 2 – Bruk av data i datasjø

6 Bruke data lagret i datasjø

Data som legges i datasjø vil i de aller fleste tilfeller være en direkte kopi av de data som ligger i kildesystemer. Dvs. at det er underlagt samme regler som kilden med tanke på sikkerhet. Rammeverket vil, som beskrevet tidligere, legge opp til en sikkerhet som er veldig streng her.

Med data i datasjø vil man få mulighet til å bruke data til ulike formål. Det er - og vil komme til - mange ulike formål. Hvert formål må ha sin egen saksgang før en eventuell implementasjon kan starte. Denne saksgangen skal sikre at det ikke trekkes ut data og gis tilgang til data som overstyrer/endrer den tilgangen som er gitt i kildesystemet.

Følgende prinsipper gjelder for all bruk av data i datasjø:

- Ingen mottakere skal ha direkte tilgang til data i datasjø
- Alle data som skal til betjeningslaget må via transformeringslaget
- For alle personopplysninger som flyttes på skal brukerloggdatabasen oppdateres med informasjon for hver enkelt person om hva som er blitt flyttet fra hvor til hvor – og for hvem.
- Mottakere kan kun aksessere data i betjeningslaget, der tilgangskontroll og andre nødvendige sikkerhetsregler er aktivert per datakilde.
- All bruk av person data skal loggføres i brukerloggdatabasen

Under følger noen eksempler på bruk/formål der data i datasjø gjøres tilgjengelig.

Eksempel	Forklaring på behov/formål	Eksempler på regler
Analyse/Rapport fra et kildesystem	Kildesystem har ikke mulighet for rapportering, eller man ønsker å	Avhengig om mottakerne er de samme som bruker kilden eller nye så kan man

	bruke andre teknikker for rapportering.	gjenbruke regler for kilden med tanke på tilgang. Forvaltning av tilganger til både data og rapporter må være håndtert.
Analyse på tvers av kildesystem	Behov for å analysere eller lage rapporter på data på tvers av mange kildesystemer.	Data må anonymiseres i transformeringen Rapporter/analyse.
Analyse med sensitive data	Behov for å analysere data fra kilder med personopplysninger.	Data lagres og transformeres gjennom egne tjenester som styres med egen sikkerhetsimplementasjon.
Innsyn	Data fra ulike systemer gjøres tilgjengelig for intern/eksterne brukere.	Innsynsløsningens saksgang håndhever regler for tilgang til data.
Uttrekk	Data gjøres tilgjengelig for interne brukere i Stavanger kommune med et tjenstlig behov der kilden ikke er sensitiv eller begrenset.	Applikasjonen(e) er ansvarlig for å håndheve regler for sikkerhet.
API	Data gjøres tilgjengelig for eksterne systemer. Azure API Gateway brukes som plattform for API.	API henter ut data fra betjeningslaget. API må ivareta sikkerhet som håndtering av tilganger.
Mobil -applikasjon	Data i datasjø gjøres tilgjengelig for en mobilapplikasjon som kan brukes av beboere i Stavanger kommune.	Data hentes ut fra datasjø via transformeringslaget. Applikasjonen må håndtere tilganger til data. Data kan ikke skrives tilbake til datasjø fra applikasjonen. Men på sikt kan mobilapplikasjonen bli en ny kilde i datasjø.
Åpne datasett	Data er definert som helt åpne og datasett gjøres tilgjengelig eksternt.	Dette gjøres gjerne i dataportaler som CKAN.

6.1 Analyse og implementasjon av bruk av data i datasjø

Hver enkelt bruk av data i datasjø må analyseres og en plan for utvikling av transformasjonsløpet av data må legges. Dette rammeverket legger ikke opp til en slik detaljert plan. Allikevel gis det her noen innspill på en del spørsmål som må inkluderes i analyse og design av løsning.

Spørsmål	**Viktig å tenke på**
Hvilke data trenger jeg og hvem eier dataene?	Data kan ikke brukes uten at det foreligger godkjenning. Behandlingsansvarlig må være involvert.
Hvilke datakilde(r) er det snakk om og er datakilde tilgjengelig i datasjøen?	Er datakilden(e) hentet opp i datasjøen, og ferdig dokumentert i datakatalogen? Hvis ikke må man kanskje ha et forprosjekt som tilgjengeliggjør alle data. Eller ta en ekstra runde på å dokumentere datakilden(e).
Hvilke tilganger/rettigheter har man i dagens kildesystem?	Har mottakerne av data tjenstlige behov? Kan prosess for tilganger gjenbrukes eller må nye implementeres? Er data underlagt lovgiving? Er data underlagt føringer for virksomhet?
Hvordan kan betjeningslaget forvalte tilgang til data?	For de aller fleste formål beskrevet over er det betjenings- og/eller presentasjonslaget sitt ansvar å videre forvalte tilgang til data. Dette må godkjennes av dataeier og analyse av sikkerhet skal gjennomføres.
Kan data anonymiseres til betjeningslaget?	Hvis data blir anonymisert, vil flere mottakere kunne få tilgang? Hva trenger man i så fall å anonymisere?

Hvordan skal data gjøres tilgjengelig for mottaker?	Hvilke alternativer har vi for å presentere data til mottakere? Er noen av mottakerne eksterne? Hvilket alternativ passer best? Bør jeg benytte mer enn ett alternativ?

7 Transformering

Transformering er første steg i prosessen der man tar i bruk data lagret i datasjø. Dette innebærer operasjoner med regler og funksjoner som utføres på data som lastes fra datasjøen til betjeningslagets lagringstjenester. Transformeringen vil variere fra situasjon til situasjon og vil bestemmes utfra behovet i betjenings- og presentasjonslaget.

7.1 Prinsipper

- Alle transformasjoner forankres i behovet i betjenings- og presentasjonslaget
- Alle tilkoblinger styres via Azure Key Vault
- Alle transformasjoner utføres i Azure Databricks
- Alle personnøkler skal pseudonymiseres gjennom transformering
- Alle personopplysninger skal anonymiseres (maskeres/aggregeres) gjennom transformering før de lander i betjeningslaget
- Hver gang personopplysninger transformeres skal det logges i brukerloggdatabasen oppdateres med informasjon om hva som har blitt gjort med den enkelte person sine data
- Begrens tilgang til Azure Databricks som vist i sikkerhetsarkitekturen. Kun autentiserte maskinkonti har tilgang
- Bruk GitHub versjonskontroll
- Sett opp tre abonnement (miljø) for datasjøen: Utvikling, test og produksjon
  og tilsvarende grener i GitHub
- Utvikling skal kun gjøres i utviklingsmiljøet

7.1.1 Orkestrering i Azure Data Factory

Azure Data Factory (ADF) brukes til oppretting, planlegging, administrering og automatisering av dataintegrasjon gjennom ADFs datasamlebånd (pipeline). For all dataoverføring som inkluderer transformasjoner, skal Azure Data Factory brukes til orkestrering og med Azure Databricks notebooks som transformasjonskomponent i datasamlebåndet. For dataoverføring som ikke krever noen transformasjoner skal det benyttes kopieringsjobber i Azure Data Factory. Dette for å redusere kostnader.

Retningslinjer for datasamlebånd i ADF:

Hva	Retningslinje	Annet
Oppretting og forvaltning	Hovedregelen for generering av datasamlebånd i Azure Data Factory er å bruke det visuelle grensesnittet i Azure. Dersom teamet som vedlikeholder løsningen ser behovet og/eller har sterk kompetanse innen utvikling av datasamlebånd utenfor det visuelle grensesnittet i Azure, kan PowerShell, Python,.NET eller ARM brukes. I så fall er anbefalt hovedregelen at kun ett av valgene benyttes for å forenkle forvaltning av løsningen.	Det visuelle grensesnittet fungerer like bra som å utvikle med tilgjengelige SDK.
Tidsplanlegging (scheduling)	Frekvensen for når datasamlebånd kjøres forankres i behovet i betjeningslaget og hvilken type data som lastes (f. eks.	Se resultat fra analysen av kildesystemet basert på *3.3.1.3 Leveranse 3 – Migrering/Integrasjon design* for

	transaksjoner, «slowly changing dimension» o.l.) Datasamlebånd planlegges for den aktuelle frekvensen via triggere. Dette gjøres i det visuelle grensesnittet. Triggere kan f. eks. være basert på at nye data har blitt lastet til en gitt katalog i datasjøen eller at et annet datasamlebånd er kjørt ferdig.	hvordan last av data skal planlegges. Last fra datasjø skal planlegges slik at det lastes etter nye data fra kilden er ferdig lastet til datasjøen.
Påloggingsinformasjon i Azure Key Vault	Bruk Azure Key Vault for alle tilkoblinger i Azure Data Factory. Dette gjør vedlikehold av løsningen mye enklere når passord endres, kompleksiteten og dermed sikkerheten til passord økes, i tillegg til at færre personer trenger kjennskap til brukernavn/passord Det må implementeres prosesser for endring av passord og Azure Key Vault må implementeres i prosessene slik at alle konsumenter som bruker aktuell påloggings-informasjon fra Azure Key Vault ikke feiler hver gang passord endres.	Alternativet er å legge inn påloggingsinformasjon i tilkoblinger i ADF hvor det lagres og krypteres. Det er mer utfordrende når passord endres, i tillegg er dette ikke like sikkert.
Datasamlebånd som inkluderer data fra aktive kildesystem	Inkluder avhengighet til last av data fra kildesystem til ADLS slik at Databricks notebooks som transformerer data fra dette kildesystemet ikke starter før ny data fra kilden er lastet.	Dersom data ikke blir lastet trigges en feilmelding.
Monitorering	Monitorer datasamlebånd for å se hvilke som har kjørt suksessfullt og hvilke som evt. feiler. Sett opp en sammendragsrapport som gir oversikt over alle datasamlebånds kjøretid og om de har kjørt suksessfullt eller feilet i løpet av den forrige perioden datasamlebåndene kjørte. Dette gjøres via «Alerts» i Azure Data Factory.	Dette kan bli automatisert via PowerShell,.NET eller Python.
Varsling ved feil	Alltid send e-post og/eller SMS dersom en aktivitet i et datasamlebånd feiler.	
Prosesser ved feil i last	Etabler prosesser for når datasamlebånd feiler slik at feilene blir tatt hånd.	
Bruk mappestruktur og navnekonvensjon	Bruk tilsvarende mappestruktur som *5.4 Katalogstruktur* for lagring av datasamlebåndene, og bruk navnekonvensjonen for navn på datasamlebåndene i ADF.	
Versjonskontroll	Alltid bruk versjonskontroll. Påse at utvikling kun foregår i utviklingsmiljøet. Mao. Brukere skal ha skriverettigheter i utviklingsmiljøet, og kun leserettigheter i testmiljøet. Produksjonsmiljøet er kun tilgjengelig for maskinkonti.	Det finnes versjonskontroll med mulighet for DevOps i ADF.
Nettleser	Bruk Microsoft Edge eller Google Chrome til arbeid i ADFs visuelle grensesnitt.	Link til dokumentasjon for Azure Data Factory

	Ifølge Microsoft er det de eneste nettleserne med full støtte for Azure Data Factory.	

I ADF blir transformasjons-aktiviteter prosessert i transformasjonskomponenter med egne beregningsressurser. Azure Databricks, Azure Functions, Mapping Dataflows og Stored Procedures er eksempel på noen av de vanligste transformasjonskomponentene i ADF.

Ved forskning på data i datasjø kan det tillates å gi tilgang til rådata i datasjøen. Det skal begrenses til kun et fåtall brukere, og vurdering og godkjenning av tilgang til data i datasjø skal innhentes fra dataeier og systemansvarlig for datasjø. Tilgang skal vurderes i hvert enkelt tilfelle og slettes etter endt oppdrag.

Det er særdeles viktig at kommunen går opp ansvarsfordeling mellom forsker og kommune rundt bruken av personopplysninger. Ved all ekstern bruk av data skal det vurderes om det er påkrevd med en databehandleravtale, personvernombud bør involveres i denne vurderingen, dette er spesielt viktig i forbindelse med forskning. Er det usikkerhet bør det kun gis tilgang til anonymiserte data som har vært gjennom en re-identifiserings risikovurdering (utført av ARX). Azure Databricks er et kraftig transformeringsverktøy som blant annet støtter fire forskjellige programmeringsspråk, kan enkelt skaleres etter ressursbehov, kan brukes til vanlig data og «big data», og kan brukes til avansert analyse og kunstig intelligens-utvikling. Delkapittelet under beskriver retningslinjer for transformering i Azure Databricks basert på nåværende beste praksis. Med tanke på at Databricks er en relativt ny tjeneste i Azure vil det stadig dukke opp forbedringer, nye egenskaper og beste praksiser som er viktig å implementere. Da skal samtidig dette dokumentet oppdateres med den nye praksisen.

7.2 Transformering

Transformasjon fra datasjølaget til betjeningslaget utføres av Azure Databricks, orkestrert av Azure Data Factory. Transformasjoner utføres ved å sette opp Databricks Notebooks som transformasjonskomponenter i Azure Data Factory datasamlebånd. Dette gir stor fleksibilitet og tillater prosessering av både strukturert og ustrukturert data, stordata, i tillegg til mulighet for varierende kompleksitet av transformasjoner.

7.2.1 Transformering i Azure Databricks

Azure Databricks skal brukes til å utføre alle transformasjoner når data lastes fra datasjø til betjeningslaget, utforske data i datasjøen og utføre avanserte analyser.

Retningslinjer for Azure Databricks:

Hva	Retningslinje	Annet
Nettleser	Azure Databricks fungerer best med Google Chrome, Firefox og Safari.	Link til Azure Databricks dokumentasjon
Oppretting av ressursen	Ved oppretting av Azure Databricks skal den knyttes til miljøets (produksjon/test/utvikling) ressursgruppe. Databricks-tjenesten skal også kobles til et eget sub-nett som inneholder Azure Storage, og nettverkssikkerhetsgruppe (NSG). Fullfør opprettingen av Databricks ved å sette opp de nødvendige sikkerhetsreglene for NSG'en til Databricks.	Azure Databricks i konfigurert virtuelt nett
Valg av programmeringsspråk	Azure Databricks tilbyr fire forskjellige språk: Spark SQL, Python, R og Scala. For transformasjoner er hovedregelen å benytte SQL for enkle transformasjoner og velge én av Python, Scala eller R for avanserte transformasjoner og analyse. Det er mest hensiktsmessig å velge det	Alle språkene blir skrevet om til Spark backend og kjører som Spark på klyngen i Databricks. Grunnen til at SQL skal brukes for enkle transformasjoner er at alle som jobber med data har et forhold til SQL.

	programmeringsspråket som flest har kjennskap til.	
Valg av klynger (knyttet til kostnad)	Cluster mode: Velg «High concurrency» for følgende tilfeller: • Flere Databricks notebooks skal kjøre parallelt • Mange utviklere skal bruke klyngen samtidig • Arbeidsoppgavene krever mye ressurser • Big data • Maskinlæring Databricks runtime versjon: benytt alltid den til enhver tid nyeste ikke-beta versjonen. Ved oppdatering til nyere runtime: Påse at den nye runtime versjonen er kompatibel med eksisterende kode i produksjonsmiljøet. Worker type: Evaluer hvilken «worker type» som skal brukes. Hovedregelen er å bruke en av «standard general purpose» typene under utvikling og skalere opp etter behov. Iterer evalueringen i testmiljøet også slik at valg av «worker type» blir mest mulig riktig til produksjonsmiljøet. Bruk overvåking til å kontinuerlig optimalisere valg av «worker type» i klyngen.	Klynge er «cluster» på engelsk. Prosesseringsevnen og prisen til klyngen øker med minne, prosessenheter, antall kjerner og evt. GPU Link til Azure Databricks priser
Start og terminering av klynger	Klynger som tidligere har kjørt og deretter blitt terminert kan startes igjen. Når de startes igjen får de den samme konfigurasjonen de hadde da de sist ble terminert, dvs. samme klynge ID, antall instanser, type instanser, IAM roller, biblioteker, osv. Ved utvikling kan klynger startes og termineres (evt. auto termineres når den ikke brukes) av utviklerne. Klynger som brukes i produksjon kan startes og termineres etter behov. Dersom det er behov for å ha klynger som kjører store deler av døgnet bør det vurderes om de skal være på kontinuerlig, da dette kan spare kostnader siden Microsoft tilbyr redusert pris for klynger som kjører over lengre perioder.	Når datasamlebånd kjøres i ADF med Databricks komponenter vil ADF starte og terminere klyngene som brukes dersom klyngen ikke kjører allerede.
Tilkobling til Azure Data Lake	Azure Databricks kobles til Azure Data Lake via Azure AD Application (AD app). Azure AD app må ha eier-rettigheter i IAM for datasjøen, og den må få leserrettigheter i mappene i mappestrukturen i datasjøen. Det må lages en AD app for hver tilgangstype mot datasjøen, f. eks. en AD app med tilgang til sensitive data og en til	Databricks kobles til en datasjømappe, og da får Databricks mulighet til å utforske innholdet i den mappen og undermappene. Dersom en datasjømappe blir tilkoblet i Databricks vil den fortsette å være tilkoblet helt til

	ikke-sensitive data. Nøkler som hører til Azure AD appene legges i hver sin Azure Key Vault.	den blir frakoblet, uavhengig av om klyngene termineres og startes.
Databricks notebooks	• Bruk kun ett programmeringsspråk i hver notebook. • Bruk cellene til å skille mellom steg i transformasjon og for å isolere evt. feilmeldinger til celler. • Legg alle Databricks notebooks som skal kjøres i et datasamlebånd i aktuelle mappe i den etablerte mappestrukturen. • Fordel dimensjonstabeller og faktatabeller i hver sine notebooks • Dersom noe skal kjøres parallelt må det splittes i to forskjellige notebooks.	
Mappestruktur for Databricks notebooks	Opprett en mappestruktur i Databricks som følger mappestrukturen i Datasjøen. Arbeidsområdet (workspace) i Azure Databricks må også ha en egen mappestruktur med navnestandard. Denne skal gjenspeile bruksområdet til filene og den etablerte navnestandarden skal brukes	
Bruk av database/tabeller i databricks-miljøet	Databaser og tabeller i Azure Databricks er lagret i DBFS (Databricks file system). Dataen ligger på Azure Storage og er dermed kryptert.	Filer i DBFS består selv om klynger termineres.
Autoskalering	• Bruk autoskalering slik at Azure Databricks tilpasser antall «arbeidere» i klyngen etter behovet • Autoskalering velger antall arbeidere mellom spesifisert minimum og maksimum ved oppretting av klyngen • Velg minimum 2 og maksimum basert på arbeidet som skal utføres av klyngen • Bruk overvåking til å finne ut om klyngens maksimum skal endres og/eller arbeidertypen skal endres • Juster opp dersom klyngen alltid kjører på maks og bruker lang tid, og juster ned dersom motsatt tilfelle	
Autoterminering	Sett autoterminering til 30 min på alle klynger som skal brukes midlertidig.	Så lenge en klynge er aktiv, selv om den ikke brukes til å kjøre noen notebooks, så koster den på Azure abonnementet. Autoterminering slår inn etter klyngen har vært inaktiv i spesifisert antall minutter.
Tilgangsstyring	• Styr tilgang i Databricks basert på roller og hvilke data hver enkelt utvikler skal ha tilgang til. Her brukes Azure Active Directory og brukere og grupper i denne kontekst.	Dette er svært viktig for å begrense tilgang til sensitiv data.

	• Begrens datatilgang gjennom begrensinger på arbeidsområde (Workspace), klynger og dbfs.	
Påloggingsinformasjon i Azure Key Vault	• Bruk Azure Key Vault for alle brukernavn, passord, «connection strings», nøkler og lignende • Bruk flere Azure Key Vault scope for å skille mellom tilganger i datasjøen • Sett opp granularitet for utviklere i Databricks slik at det får den tilgangen de trenger og ikke får tilgang til de scopene, og dermed data, de ikke skal ha tilgang til	Alternativet til Azure Key Vault er å hardkode, noe som både fører til feil når passord og brukernavn endres og går ut over sikkerheten. Dette er ikke et reelt alternativ, og *skal aldri* benyttes.
Dataanalyse	Dataanalysen som beskrevet i *6 Bruke data lagret i datasjø* og som ble gjennomført i forbindelse med kopiering av kildesystem til datasjø, gir innsikt i hvilke data som skal transformeres	
Anonymisering	Alle sensitive data - som ikke har et særskilt behov for å vises som de er - skal anonymiseres. Dette kan blant annet gjøres ved å skifte dato, fjerne kolonner, aggregere eller lignende.	Se utdypende beskrivelse i kapittel 14 (anonymisering).
Logging	Alle personopplysninger som blir behandlet i Databricks skal loggføres i brukerloggdatabasen med hva som er behandlet om hvem og for hva og til hvor.	
Kopieringsjobber	Dersom det ikke er behov for noen transformering fra datasjø til lagringsdestinasjon i betjeningslaget kan Azure Data Factory brukes uten Azure Databricks. Da settes det opp en kopieringsjobb.	
Transformering	Transformeringen skal gjøres med utgangspunkt i behovet i betjeningslaget. I tillegg må det tas stilling til følgende: • Masterdata • Datavasking • Datakvalitet (datoformat, nummer som er desimal, int eller string) • Tabellnavn, Begrepsnavn og kolonnenavn	Se også *6 Bruke data lagret i datasjø*, for eksempel på bruk av data i datasjø og transformeringer.
Databricks plan	Bruk Premium Plan for produksjonsmiljøet. Dette tillater blant annet granularitet i tilgang til Azure Key Vault, det kan velges blant flere «worker types» og mulighet for støtte fra Azure support. For utvikling/test kan standard plan for Databricks brukes så lenge alle utviklere som har tilgang til Databricks også skal ha mulighet til å lese alle filer i utvikling/test for datasjøen	
Versjonskontroll	Konfigurer GITHub for Databricks og sett opp et DevOps miljø.	Merk at Databricks har sin egen versjonskontroll i tillegg.

Virtuelle nett	Ved oppretting av Databricks ressurs skal denne tilegnes det aktuelle (produksjon/test/utvikling) virtuelle nettet. Det gjøres ved å spesifisere et privat og offentlig sub-nett innenfor IP-rekkevidden til det aktuelle virtuelle nettet	Merk at Databricks-ressursen setter opp sitt eget virtuelle nett i tillegg til dette.

7.2.2 Transformering i Stream Analytics

Azure Stream Analytics skal brukes når data fra Tingenes Internett (IoT) skal transformeres, det være seg direkte til betjeningslaget eller for lagring av rådata.

Retningslinjer for Azure Stream Analytics:

Hva	Retningslinje	Annet
Verktøy	Stream Analytics kan enten utvikles direkte i Azure portalen, eller i Visual Studio med tilleggskomponent for Stream Analytics og Data Lake Tools.	Link til Azure Stream Analytics dokumentasjon
Oppretting av ressurs	Ved oppretting av Stream Analytics-jobb skal den knyttes til rett ressursgruppe (utvikling/test/produksjon).	
Skalering og monitorering	Så sant ikke analyse av kildedata indikerer størrelse på datastrømmen (antall meldinger per sekund eller minutt) bør første oppsett av tjenesten legges på standardnivå som er 3 STU.	Det er fullt mulig å skalere tjenesten ned til 1 STU hvis man over tid ser at kapasiteten ikke utnyttes riktig.
Sikkerhet	Stream Analytics skal ligge i transformasjons-subnettet og ha samme sikkerhetsgrupper knyttet til seg som resten av laget.	
Kildekontroll	Det anbefales å utvikle Stream Analytics-jobber i Visual Studio, da det gjør det mulig å lagre koden i GitHub. I skrivende stund er eneste grunnen til å ikke benytte Visual Studio hvis man må levere sanntidsdata direkte til Power BI, da denne typen leveranse ikke er støttet i Visual Studio ennå.	
Schema on read or write	Alle data som transformeres direkte til betjeningslaget skal benytte «Schema on write». Data som skal lagres som rådata bør benytte «Schema on read» for fleksibilitet.	
Anonymisering	Alle sensitive data skal anonymiseres når tjenesten skriver til betjeningslaget. For rådata i sikker sone er ikke dette nødvendig.	Se kapittel 14 for detaljer rundt anonymisering.
Transformering	Avhengig av behov og bruksområde gjøres transformasjon av sanntidsdata i Stream Analytics når data sendes til betjeningslaget eller direkte til visualisering (Power BI)	

7.3 Overvåking og analyse av transformasjons-jobber

Azure Data Factory tilbyr et visuelt verktøy for å utforske om datasamlebåndene har kjørt gjennom ok, eller om noen av jobbene har feilet, og evt. hvilken aktivitet som har feilet. Dette skal brukes aktivt for å påse at alle datasamlebånd som settes opp kjører transformasjonsjobben vellykket gjennom til lagringsdestinasjonen sin. Alle datasamlebånd som feiler, skal feilsøkes og rettes opp i så fort det lar seg gjøre. Feil og løsninger skal

loggføres for senere oppslag og for læring for hele teamet. Dette er en del av den kontinuerlige forbedringen av datasamlebåndene som.

I tillegg til skal alarmer i Azure Data Factory brukes til å sende en sammendragsrapport for jobber som har kjørt suksessfullt, feilet og hvor lang tid hver jobb brukte. Dette kan brukes i kontinuerlig forbedring, både for å fikse Transformasjonsjobber som feiler i tillegg til å få oversikt over Transformasjonsjobber som bruker lang tid og sette i gang arbeid for å effektivisere de jobbene.

7.4 Sikkerhet

I Azure Databricks skal det settes opp sikkerhet knyttet til prosessering, kode og data. Dette gjøres på brukernivå gjennom autentisering og tilgangsstyring for mottakere, arbeidsområde, klynger, Azure Key Vault, DBFS og jobber. Dette sørger for sikkerhet for alt som foregår internt i Azure Databricks. I tillegg skal koblingen til datasjølaget og betjeningslaget sikres. Dette gjøres ved å sette opp et eget virtuelt nettverk og NSG for Transformeringslaget, og når Databricks-ressursen opprettes tilegnes den til det virtuelle nettet med brannmur. Da velges det samtidig et offentlig sub-nett som brukes til kommunikasjon med Azure Databricks kontroll planet, og et privat sub-nett for kommunikasjon internt i en klynge. Dette oppsettet gjør det mulig å spesifisere IP-adresser som Databricks skal utveksle data med, og på den måten kun tillater kommunikasjon med datasjø-laget og betjeningslaget slik som arkitekturen legger opp til.

Filer, tabeller og databaser i Databricks ligger lagret i Azure Storage knyttet til Databricks ressursen, og dermed vil filene alltid være kryptert.

Azure Key Vault skal brukes i alle ledd i forbindelse med transformasjonsjobbene, både i Azure Data Factory ved tilkobling til kilder, og i koden i Databricks notebooks. Det betyr at alle tilkoblinger til kilder, lagringsdestinasjoner i Azure, nøkler til Azure komponenter, brukernavn og passord skal lagres i Azure Key Vault. Det må lages like mange Azure Key Vault som det finnes tilgangsstyringer i Azure Databricks og Azure Data Factory. For eksempel trengs det en Azure Key Vault som inneholder tilganger til ikke-sensitive data og en annen for tilgang til sensitive data, på den måten begrenses utvikleres tilgang til datasjø og betjeningslag.

7.5 Last til Azure lagringstjenester

Azure Lagringstjeneste	**Tilkobling**
Azure SQL DB	Bruk JDBC tilkobling. Merk at denne koblingen har begrenset dataoverføringshastighet
Azure Data Lake store	Koble Azure Databricks til mappe med Azure AD applikasjonen. Denne koblingen har høy dataoverføringshastighet.
Azure Blob Storage	Koble Azure Databricks til mappe med Azure AD applikasjonen.
Azure Synapse SQL Pools	Data lagres til en felles Azure Blob Store som brukes av Azure Databricks og SQL Pools

Merk at JDBC koblingen fra Databricks til database ikke har så høy overføringshastighet. Som konsekvens av dette bør ikke Azure Databricks brukes til å laste datamengder større enn 5-10 GB til eller fra database. I stedet skal data lagres som fil i ADLS. I den konseptuelle datasjøen kan det settes opp en kopieringsjobb i ADF for last fra databasen til ADLS som Databricks kan hente fra, og i betjeningslaget skal Databricks lagre filen til ADLS I betjeningslaget og deretter kopierer ADF data videre til database. Dette vil spare kostnader siden Databricks-klyngen kan termineres tidligere og gjøre lasten av data mye raskere siden ADF laster data parallelt.

Når ADLS brukes som mellomlager på denne måten skal det etableres tilgangsstyring til den mellomlagrede dataen som likt eller strengere enn tilgangen til databasen dataen ble lastet fra. Så lenge det ikke er behov for å beholde den mellomlagrede dataen skal den slettes etter den er lastet videre.

8 Betjening

Denne seksjonen dekker betjeningslaget. Her vil data bli tilpasset de ulike behovene/formålene slik at det kan bli tilgjengelige til mottakere via visualiseringslaget.

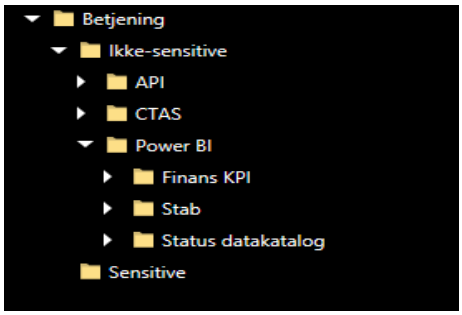
For noen eksempler på bruk se kapittel *6, Bruke data lagret i datasjø*.

Videre vil dette kapittelet kun gå i detaljer på data som brukes til analyse.

8.1 Systemer

Azure SQL DB er standard relasjonsdatabase, og her vil man kunne legge ferdig transformerte data i tabeller. SQL DB støtter flere ulike sikkerhetsmekanismer som gjør den egnet til å betjene både sensitive og ikke-sensitive data. Data lagret i SQL DB kan enten lagres i form av «data marts» med dimensjoner og fakta, eller i dedikerte tabeller som ikke er knyttet til en data mart. SQL DB støtter radnivå-sikkerhet på tabeller og dynamisk datamaskering. Dette betyr at det er mulig å legge sensitive data i denne tjenesten.

Azure Storage er kjernekomponenten i datasjøen for lagring av rådata. Det er også fullt mulig å lagre transformerte data i Data Lake Store som en del av betjeningslaget. Både fordi dette er billigere enn SQL DB, men også fordi en relasjonsdatabase ikke alltid er nødvendig – for eksempel der alle data er lagret i en enkelt fil. Det er viktig at man i bruken av Data Lake Store benytter den instansen av systemet som er koblet til betjeningslaget og ikke lagringslaget. Dette på grunn av de sub-nett og sikkerhetsgrupper som ligger på de ulike lagene.



Azure Synapse Analytics – SQL Pools er en relasjonsdatabase for store datamengder (>1TB) og spesialdesignet for datavarehusdata i stjerneskjema. Dette er en forholdsvis dyr tjeneste, men i forbindelse med Microsofts satsning på Azure Synapse Analytics som et felles analyseverktøy hvor Azure Data Factory, Databricks (Azure Spark), SQL Pools og Power BI får et felles utviklingsmiljø kan det være aktuelt å ta i bruk disse tjenestene i løpet av et år eller to.

Sikkerhets-API er et spesialutviklet grensesnitt som ligger mellom datalagringstjenestene i betjeningslaget og visualiseringslaget. Dette vil ha en rekke forretningsregler som gjør det mulig å analysere om data kan utleveres dekryptert eller ikke, om datasettet har for dårlig anonymiserings-score til å utleveres eller ikke.

8.2 Definisjoner

Katalogstruktur for Azure Storage i betjeningslaget

Hovedkatalog: Betjening

Underkataloger nivå 1: Sensitive og Ikke-sensitive

Underkataloger nivå 2: Visualiseringssystem

Underkataloger nivå 3: Konsept

Navnestandard og struktur i relasjonsdatabaser:

- Alle tabeller skal navngis etter hva slags data de inneholder
- Alle tabeller skal tilhøre et skjema
- Skjemaet knyttes til det systemet i visualiseringslaget som skal hente data fra tabellene

8.3 Prinsipper

- All lesing av data fra datasjø til betjeningslag skal gå via transformasjonslaget
- All lesing av data skal være autentisert
- All tilgang til data skal være styrt av roller
- All lesing av data skal loggføres
- Alle personopplysninger skal som standard være pseudonymisert eller anonymisert
- Alle personopplysninger som ikke kan være anonymisert skal beskyttes med kryptering
- Alle andre data klassifisert som begrenset eller sensitive skal behandles etter samme prinsipper
- Dekryptering av krypterte personopplysninger gjøres i egne sikkerhets-API som bruker tilgangskontroll for å verifisere om krypterte data kan leveres ukryptert
- Utlevering av anonymiserte data gjøres via samme sikkerhets-API som bruker re-identifiserings score på datasett for å validere utlevering der det trengs
- All lesing av personopplysninger skal loggføres i brukerloggdatabasen
- Betjeningslaget skal aldri skrive data tilbake til lagringslaget i datasjø

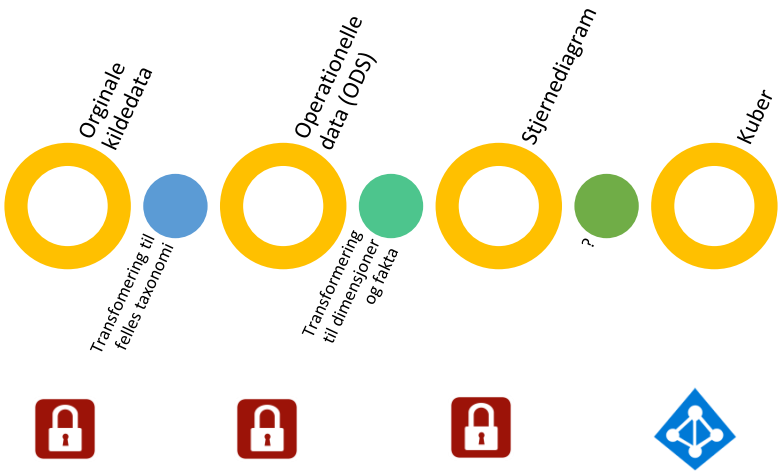
8.4 Sikkerhet

Data som blir gjort tilgjengelig i betjeningslaget kan gå igjennom flere steg før det blir gjort tilgjengelig for visualisering. Det skal være tydelig definert hvilke datakilder i dette laget som skal kunne brukes hvordan i et presentasjonslag. Altså, hvilke datakilder inneholder anonymiserte data og hvilke inneholder krypterte data.

Det skal være separate database-tjenere for ikke-sensitive og sensitive data – altså krypterte og/eller anonymiserte data. Sensitive datakilder må kobles via sikkerhets-API, mens ikke-sensitive datakilder kan kobles til direkte.

Når det er definert hvilken kilde som åpner opp for brukere skal de sikres via Active Directory og roller. Her må det velges om eksisterende grupper og rolle skal gjenbrukes eller om det må defineres nye.

Et eksempel på dataflyt i betjeningslaget



9 Visualisering/Presentasjonslag

Denne seksjonen dekker visualisering og presentasjon av data til de ulike mottakerne. For definisjon av mottakere, se kapittel 2.3, *Roller*.

Siden visualisering/presentasjon kan komme i mange ulike former vil dette rammeverket kun legge til grunn en del generelle regler. For noen eksempler på bruk se kapittel *6, Bruke data lagret i datasjø*.

9.1 Analyse

Analyse av data fra datasjø både som direkte rapporter på et gitt kildesystem og på tvers av flere kildesystemer er et kjent behov i Stavanger kommune og en av de viktigste formålene med å legge data i datasjø.

9.1.1 Prinsipper

- All tilgang til **data** skal legges på betjeningslaget via AD grupper
- All tilgang til **rapport** skal være autentisert
- Alle offisielle rapporter skal gjennomgå et utviklingsløp (Krav/behov, design/utvikling, Testing, Publisering/Produksjon) med en godkjennelses steg før publisering
- Alle uoffisielle rapporter skal kunne dokumentere at de ikke benytter sensitive data

9.1.2 Definisjoner

Begrep / Definisjon	Forklaring
Offisielle rapporter	Kjennetegn offisielle rapporter: • Brukes av mange • Bruker kun IT-kontrollerte rapporteringskilder • Bruker kun godkjente rapporteringskilder • Rapporten har en definert eier • Kan være tilgjengelig eksternt • Har blitt testet og godkjent av rapporteier • IT har forvaltningsansvar (inkluderer publisering) for konsernrapportering
Uoffisielle rapporter	Dette er rapporter som ikke er ment delt med andre eller kun i et begrenset omfang. Typisk for prototyping eller eget bruk. Kjennetegn uoffisielle rapporter: • Brukes av få • Bruker potensielt ikke godkjente datakilder • Bruker potensielt ikke IT-kontrollerte datakilder (les: datasjø) • Har ingen definert eier • Mangelfull sikkerhetsdefinisjon • Har ikke blitt testet og godkjent (les: vært del av et utviklingsløp) • Rapportutvikler har forvaltningsansvar
Rapporteringskilde	Data som er blitt gjort tilgjengelig i betjeningslaget.
Eksterne kilder	Det kan være at visualiseringslaget henter data fra andre kilder enn det som dekkes av dette rammeverket.
Power BI **(Power BI Service)**	Microsoft sin skytjeneste for rapporter.

9.1.3 Sikkerhet

Sikkerhet og tilgang til rapporter og data, kontrolleres og administreres i flere dimensjoner. Som et grunnleggende prinsipp skal alle data beskyttes ved rapporteringskilden. I dette tilfellet vil det være i betjeningslaget.

De ulike nivåene for sikkerhet blir da som følger.

- Datasikkerhet – Data beskyttes på rapporteringskilden, dvs. betjeningslaget
    - For offisielle rapporter vil dette bety at ingen data blir gjort tilgjengelig uten en formell godkjenning for den enkelte bruker.
- Publisere data – Et begrenset antall brukere vil ha lov å publisere offisielle rapporter. Flere utviklere av rapporter vil få mulighet til å publisere uoffisielle rapporter.

- Rapporter – Brukere vil få tilgang til en eller flere rapporter som vil være styrt av Sikkerhets- eller global administrator. I en del tilfeller vil administrator delegere forvaltningsrett til rapporter til områdeeier.

Mottakere:

Mottaksgrupper	Beskrivelse
Anonyme brukere	Mottaker av data er ukjent. Data som legges ut på internett vil ha mottaker som ikke er autentisert. Mao er det kun rapporter som inneholder data som er definert som åpne som vil bli tilgjengelige her.
Autoriserte brukere – Eksterne	Stavanger kommune kan velge å dele data med eksterne brukere eller eksterne systemer. Ulike eksterne brukere kan ha ulike tilganger.
Autoriserte brukere – Interne	Dette er alle brukere som blir definert som interne for Stavanger kommune. Dette kan også være innleide kontraktører.

Videre tilganger til data vil kunne bli gitt basert på en brukers tilhørighet til de ulike fagområder. Noen eksempler under.

Videre inndelinger	Beskrivelse
Avdeling /Gruppe	Brukere kan tilhøre en avdeling som håndterer og behandler en type data.
Rolle / Stilling	Basert på en brukers rolle/stilling kan data bli gjort tilgjengelig.
Geografisk lokasjon	Basert på hvor bruker er lokalisert skal bruker kun se data relevant for denne lokasjonen.

Prosessen/Prosedyren for å gi tilgang til data er IKKE dekket av dette dokumentet. Stavanger kommune må sikre at eksisterende prosesser/prosedyrer dekker tilgang til visualisering, eventuelt lage nye prosedyrer for å dekke dette.

9.1.4 Verktøy for visualisering (PowerBI og Excel)

Det finnes flere verktøy for presentasjon av data til mottakere, men her er to vanlige tjenester fra Microsoft for å eksemplifisere tilgangen. Disse er allerede i bruk i Stavanger kommune.

9.1.4.1 Excel

Excel er med sin Power Query og Power Pivot funksjonalitet en av de aller enkleste presentasjonsverktøyene man har. Dette er ofte standard programvare på PCer og mange brukere kjenner til verktøyet.

For en introduksjon til komponentene ser her.

Excel kan koble seg mot betjeningslaget eller mot andre datasett, for eksempel publiserte Power BI datasett. Excel vil alltid være en on-premises løsning selv om den kan være tilgjengelig via OneDrive.

Excel filer vil ikke direkte kunne styre tilgang til rapportene, men via tilgang til disk lokasjoner kan man simulere tilgang. Tilgang til data skal styres i betjeningslaget. Vær oppmerksom på at når Excel lagrer ned filen vil det ligge data der fra forrige bruker som neste bruker ikke skal ha tilgang til.

Excel er ikke anbefalt brukt i stor skala, men kan fungere greit for en liten gruppe. Pris kan også være en faktor da Excel ikke koster noe ekstra.

9.1.4.2 Power BI

Power BI er et veldig kraftig verktøy for å trekke ut, transformere, analysere og presenter data.

All dokumentasjon for Power BI ligger her.

Power BI består av flere områder

- Power Query – lese, vaske, rense og transformere data
- Power Pivot - Datamodellering

- Power BI tjeneste - Dele og samhandle med andre
    - Web, mobil
    - Integrere til andre komponenter i Office 365, f.eks. Flow

For offisielle rapporter som publiseres skal de to første punktene over legges i betjeningslaget. Det er kun under utvikling og forskning at man skal tillate denne bruken av Power BI.

Det finnes to muligheter for å publisere Power BI, Power BI tjeneste(sky) og Power BI Reporting Server (on-premises). Stavanger kommune benytter kun Power BI tjenesten.

9.1.5 Regler/Anbefalinger for bruk av visualisering verktøy og plattformer

Reglene her gjelder for der verktøyene er satt i system og publisert til potensielt mange mottakere. Andre brukere som bruker verktøyene til utforsking og analyse faller i en annen kategori. De vil har særskilte rettigheter til data. Mottakere her vil være de som bruker verktøyene kanskje daglig for å ta gode beslutninger.

Regel	Beskrivelse	Hvorfor
Ingen transformering av data skal skje i front-end	Ofte er presentasjonsverktøyene svært kraftige på å transformere data. Dette er allikevel jobb som i størst mulig grad skal være betjeningslaget.	Mulighet for gjenbruk av transformasjonsregler til mange ulike rapporter. Bedre fleksibilitet i valg av presentasjonsverktøy da data er 'ferdig'.
Bruker-ID må kunne sendes til betjeningslaget	Tilgang til data skal ikke gjøres i presentasjonslaget selv om dette er mulig.	Datasikkerhet skal ivaretas i betjeningslaget der det vil være enklere å vedlikeholde via AD grupper. Både Excel og Power BI.
Brukere av rapporter (og datasett) skal være autentisert		Ved å skille på tilgang til rapporter og data vil man få bedre fleksibilitet. Dette vil typisk bli vedlikehold av ulike roller.
Alle rapporter/instrumentbord skal ha en eier	Rapporter bør organiseres. Typisk under de ulike fagområder. Disse igjen vil ha en definert eier. All utvikling som gjøres er basert på behov fra forretningen. Det er viktig at de er involvert og har eierskap til det som blir produsert.	
Offisielle rapporter skal gå igjennom et utviklingsløp	Ved å gå igjennom en strukturert behov/analyse/design, utvikling, testing og publisering vil kvalitet og sikkerhetsregler bli ivaretatt.	Man unngår at rapporter publiseres før de er ferdig der man ikke har testet kvalitet og sikkerhet.
Uoffisielle rapporter skal IKKE deles med andre	Rapporter som er utviklet i forbindelse med forskning skal ikke publiseres/deles i offisielle kanaler.	
Eksterne kilder skal ikke brukes	Eksterne kilder skal ikke brukes direkte i visualiseringslaget. Hvis det er behov for å hente kilder eksternt skal de først inn i datasjø (eller i betjeningslaget).	Dette sikrer at eksterne data er dokumenter og brukes riktig.
Personopplysninger kan i visse tilfeller deles i en rapport	Ved forretningsmessig behov, og etter godkjenning kan	Dette må ha et eget godkjenningsløp der

	personopplysninger deles i en rapport. For eksempel i et HR rapporterings-system.	personvernombudet skal godkjenne bruken før rapporten utarbeides.
Rapporter med personopplysninger skal overvåkes særskilt	All bruk/konsum av personopplysninger skal føres i brukerloggdatabasen, der det oppdateres hvilke personers data som er blitt vist, hvor, når og til hvem.	

9.2 Applikasjoner/Tjenester

Her er det nesten uendelig mange muligheter for hva data kan brukes til og som ikke er analyse. Noen av disse er beskrevet i kapittel *6, Bruke data lagret i datasjø*.

Definisjon på applikasjoner i denne konteksten er alle applikasjoner (inkludert tjenester) som henter data fra betjeningslaget av datasjøen. Det vil finnes applikasjoner/tjenester som i natur er like de som henter data fra datasjø, men de er ikke inkludert i denne definisjonen..

Applikasjoner vil typisk ha et brukergrensesnitt som er ment for sluttbrukere, mens tjenester vil typisk ha et programmeringsgrensesnitt som andre applikasjoner kan bruke for å få tilgang til data.

9.2.1 Prinsipper

- Applikasjon er ansvarlig for tilgangsstyring til data i henhold til gjeldende regler for datasjø og kildesystem der data oppstod
- Applikasjonen er ansvarlig for autentisering av mottaker
- Applikasjoner som skal ha tilgang til personopplysninger må oppdatere brukerloggdatabasen med informasjon om all bruk av data på personnivå

9.2.2 Data

En applikasjon vil som beskrevet få levert data fra datasjø. I tillegg antas det at det ofte også er et sett av data som eies og administreres av applikasjonen. De er det helt og holdent applikasjonen selv som håndterer.

Data som ikke allerede ligger i datasjøen kan da vurderes å legges til i datasjøen. Hvis man ønsker å lage løsninger for analyse/rapportering på disse dataene, må de legges i datasjøen sammen med andre data slik at de følger samme regler og løsninger. Duplisering av data skal unngås der løsningen allerede har hentet data fra datasjø.

Der sensitive data fra datasjø blir delt uten å bli anonymisert skal data kategoriseres likt som i kilden.

9.2.3 Tilganger

Applikasjoner kan få tilgang til data fra datasjø (via transformasjon) hvor det allerede finnes regler for tilgang. Disse må videreføres slik at ikke data blir gjort tilgjengelig for mottakere som ikke skal ha tilgang.

Applikasjoner kan få sin egen behandlingsansvarlig som ikke er den samme som eier data i kilde/datasjø. Da må det defineres tydelig hvem som har ansvar for å gi tillatelse til å gi tilganger.

Del 3 – Felles funksjoner

10 Overvåking

Rammeverket legger opp til en hovedregel der alle komponenter som er i bruk i Azure skal skru på logging slik at man kan overvåke Azure komponentene. Ved å etablere hensiktsmessig logging kan en benytte dette til å oppdage, forstå, og/eller gjenopprette etter et angrep, internt eller eksternt.

Målet med loggingen er at den samler inn sentral informasjon om komponenten, slik at man får en forståelse for hver komponents bruk og slik at man kan definere KPI'er og krav til loggene. Da kan man skru på alarmer eller lage rapporter/uttrekk på dataene som ligger i loggene. Flere av komponentene har allerede Dashboards osv. som man kan bygge videre på.

Prinsipper for overvåking er:

- Alle Azure komponenter skal logge all aksessering av komponenten
- Hvis man skrur på logging må det være en ide om hva som skal overvåkes.
- Alle data aksess skal logges, dette inkludere også aksess til data som blir avvist
- All trafikk til en komponent skal logges?

Ved å overvåke kan man å oppnå følgende

- Få oversikt over mottakere som bruker de ulike komponentene
- Gi alarmer/målinger ved unormal aktivitet
- Få oversikt over alle data aksess, spesielt til sensitive data
- Få oversikt over hvilke data i datasjø som er i bruk
- Hvilke data den enkelt mottaker har tilgang til
- Identifisere kritiske punkter for ytelse, ytelsesforbedring eller optimalisering

Det er ikke sikkert at alt som nevnes her fås direkte fra dagens tjenester i Azure. Stavanger kommune må sikre at logger/alarmer blir gjort tilgjengelig for den gruppen av mottakere som trenger dette. Det kan være internt i IT, men også til systemansvarlige/behandlingsansvarlige. Neste underkapittel sier noe om dette.

Azure Monitor og Azure Security Center benyttes her som logganalyseverktøy og vil varsle dersom det oppdager unormal aktivitet. Sikkerhets- og global administrator må legge opp rutinene for overvåking og aksjoner. Det inkluderer definering av alarmer, hvem som skal varsles og hvordan – samt hvem som skal overvåke instrumentbordene i Security Center – og hvordan dette skal gjøres. Samt hvilke handlinger som må utføres ved en alarm.

Logger kan inneholde sensitiv informasjon om den enkelte medarbeider og behovet for lagring av slik informasjon må til enhver tid veies opp mot behovet for personvern. For mer informasjon se Nasjonal sikkerhetsmyndighet (NSM) her.

Stavanger kommunes sikkerhetsadministrator og global administrator må utarbeide retningslinjer for overvåking og oppfølging.

10.1 Rapporter og analyse rundt implementasjon av rammeverk

Det kan være svært hensiktsmessig å utvikle rapporter og verktøy som analyserer og sjekker riktig bruk av komponentene til Azure. Dette vil være i tillegg til det som kommer som en del av plattformen og som beskrevet over. Løsninger som dette vil tilpasses Stavanger kommune sine egne regler og vil da gi mulighet for kontroll på etterlevelse av disse interne reglene.

Det er sterkt anbefalt at dette gjøres helt fra starten og videreutvikles ved behov underveis som man ser behovene. Dette vil framstå som en ekstra kost, men innsikten slike rapporter skaper vil over tid gi en veldig god kontroll på hva som finnes og gjøres av de ulike prosjektene som vil lever framover. Det er forventet at mengden data og bruksområder vil øke dramatisk over de neste årene.

Mottakerne av slike rapporter vil typisk ikke ha tilgang til Azure eller svært begrenset tilgang. Uansett vil målet med slike rapporter være å skape oversikt og kontroll av det som finnes på Azure plattformen.

Noen eksempler

- Man følger rammeverkets regler for sikkerhet
- At man oppdager gap/mangler så tidlig som mulig
- Gir oversikt til ulike brukergrupper på hva som er implementert i Azure

Eksempler på slike rapporter

- Alle AD kontoer med tilhørende adganger til tjenester og data
- Bruk av datakatalog og kontroll på til enhver tid gjeldende regler
- Bruk av data kilder igjennom systemet slik at behandlingsansvarlige vet hva data brukes til.
- Dokumentasjon av data, rapporter, kilder osv. i Datakatalog
- Samlet oversikt over alle Data Factory jobber og deres tilstand

11 Brukerloggdatabase

Neo4J – eller Cosmos DB med Graph API – skal brukes for å loggføre alle hendelser knyttet til personopplysninger i datasjøen. Fra første inntak av data til lagring, transformasjon og servering. Hvert eneste tilfelle hvor persondata er brukt skal inn i brukerloggdatabasen. Denne er nøkkelen til å kunne gi svar på alle henvendelser knyttet til utøvelse av datasubjektets rettigheter i personvernlovgivningen.

Hvis persondata blir anonymisert skal brukerloggdatabasen bli oppdatert med en rad om at data er anonymisert i et datasett. Det vil da ikke være mulig å videre dokumentere bruk av data på et slikt datasett på personnivå.

12 Pseudonymisering

«Pseudonymisering» i dette tilfellet er maskering av personopplysninger slik at de ikke kan identifiseres uten bruk av tilleggsopplysninger. Det er en forutsetning at nevnte tilleggsopplysninger lagres atskilt og omfattes av tekniske og organisatoriske tiltak som sikrer at personopplysningene ikke kan knyttes til en identifisert eller identifiserbar fysisk person. (GDPR art 4.5)

Pseudonymisering av personopplysninger skal skje i forbindelse med dataimporten til datasjøen. Dette gjøres ved at nøkkelen til personidentitet byttes ut med en Pseudo ID. Vanligvis er denne nøkkelen personnummeret, men det kan også for eksempel være ansattnummer eller pasientnummer.

Data om samme person fra forskjellige kilder skal ha forskjellig Pseudo ID. Den virkelige identiteten og Pseudo ID lagres i Hashicorp Vault slik at man kan finne til bake til riktig person og samle dataene i analyser på en sikker måte.

13 Hashicorp Vault – nøkkelhåndtering

Hashicorp Vault skal brukes til å håndtere alle nøkler relatert til sensitive data og all kryptering av datafiler i datasjøen.

Ved pseudonymisering er det også Hashicorp Vault som håndterer utdeling av nye pseudo ID og sikker lagring av de orginale ID’ene.

14 Anonymisering

Personopplysninger som leveres ut utenfor de spesifikke tilfellene hvor det er nødvendig utfra et spesifikt behov – som for eksempel i enkelte HR rapporter – skal anonymiseres. Anonymisering er definert som teknikker for å innføre støy i datasettet slik at det ikke er mulig å re-identifisere personer i det anonymiserte datasettet.

Nøyaktig hvilke anonymiseringsteknikker som må benyttes vil variere fra datasett til datasett, men typiske teknikker er

- Skifting – for eksempel flyttes fødselsdato til den 15 i hver måned, eller i kvartalet
- Avidentifisering – flest mulig sensitive kolonner fjernes fra et datasett
- Geografisk gruppering – bruk postnummer eller kommunenummer heller enn gateadresser
- Aggregering – vis summerte tall i stedet for enkeltrader
- Avgrensning – ikke vis datasett med mindre enn X antall rader

Det er vanlig å kombinere minst to av disse.

Verktøyet ARX skal kjøres på alle anonymiserte datasett for å skape en re-identifiserings risikovurdering. Denne vurderingen kommer i form av et tall, som så skal brukes for å vurdere hvor høy risiko det vil være for å utlevere et gitt anonymisert datasett. ARX risiko score skal inn som en verdi i alle analyserte datasett, da dette er en verdi som sikkerhets-APIet benytter i analysen av hvilke data som kan vises.

Alle datasett med personopplysninger skal gjennom en analyse hvor beste anonymiseringsmetode vurderes.

Man skal velge kraftigst mulig anonymisering mulig for datasettet. Så kan man skifte en dato til midt i kvartalet uten at det ødelegger for analyser så skal det gjøres. Alle sensitive kolonner som ikke er nødvendige for å utføre analyse skal fjernes. Dukker det senere opp et behov som krever en kolonne som ikke er med i det eksisterende datasettet så må datasettet gjennom en ny analyse for anonymisering. Kolonner skal altså ikke med utfra et «kjekt å ha» prinsipp, kun etter identifisert behov.

Avgrensning av datasett gjøres typisk i sikkerhets-APIet, da kombinasjon av flere datasett kan medføre økt risiko. Selv om det jevnt over vil være mulig å fastsette en standard på minimum rader i et datasett før det kan utleveres, så vil man på et generelt grunnlag kunne si at det er anonymt hvis det er minst 5 personer (K-anonymitet) kombinasjonen av data referer til, det vil i mange tilfeller være behov for å sette grensen høyere, dette avhenger av hvor stort datagrunnlaget er og hvilke data det gjelder.

> Eksempel: hvis dataene relaterer seg til hvilke bilmerket kvinner og menn som bor i et område bruker når de kjører til jobb (data: kjønn, bilmerke og postnummer), så må det være minst fem personer (K-anonymitet) som har det samme kjønnet, bilmerke og postnummer, for at det skal være anonymt.

Nivået som kreves for å anonymisere avdekkes i analysefasen og gjennom bruk av ARX risikovurdering, dette må vurderes i hvert tilfelle.

En god guide til anonymisering fra det engelske datatilsynet: https://ico.org.uk/media/1061/anonymisation-code.pdf

15 Datakatalog

Azure Data Catalog skal brukes til å dokumentere alle kilder i Stavanger kommune. Formålet med dokumentasjonen er at man skal få oversikt av hva som finnes av data i ulike systemer slik at man vet hva som eksistere og hvor det er. Det skal ikke være tvil om hvilke data man har og hvordan de brukes.

Det vil ta tid, men hovedregelen vil være at følgende skal dokumenters

- Kildesystemer som også eksisterer i datasjø

    - Kildesystemer som bare finnes i datasjø
- Datakilder i betjeningslaget
- Rapporter og datasett

15.1 Prinsipper

- Datakatalogen er hovedlageret for alle metadata i organisasjonen
- En datakatalog inneholder både kilder som kun et begrenset antall brukere skal se, og kilder som er tilgjengelig for alle.
- Alle kilder som er i aktiv bruk i organisasjonen bør katalogiseres og vedlikeholdes
- Rådata i datasjø der data er like som i kilden skal ikke dokumenters
- Data som ligger i betjeningslaget i datasjø skal dokumenters
- Alle kilder skal ha en ekspert som kan vedlikeholde informasjonen
    - Denne eksperten bør være forretningsbruker som har inngående kjennskap til systemet kilden er koblet til.
- I de tilfeller der kilden har tabeller uten data (typisk for ERP systemer) – og er aktive, bør selv tomme tabeller katalogiseres – men merkes som tomme.
- Kilder i datasjø som er inaktive trenger kun å dokumentere objekter som er relevante

15.2 Roller og tilganger

Roller kan deles i tre:

- Datakatalog Administrator – Styrer hele katalogen med alle brukere og kilder. Ansvarlig for å retningslinjer på dokumentasjon og kontrollere at disse blir fulgt
- Behandlingsansvarlig – Ekspert på kildesystem og har ansvaret for at det er riktig dokumenter i data katalog.
- Bidragsyter – Bidrar til a kilden er riktig dokumenter i data katalog

I tillegg har katalogen gi lesetilgang til andre brukere som ønsker informasjon om de ulike kildene

Tilgang til å vise data via Datakatalog skal ikke være mulig da det kan føre til brudd på sikkerhetsregler

15.3 Taksonomi og vokabular

Stavanger kommune må definere et felles vokabular (Glossary Term) for å best å kunne dokumentere alle kilder på tvers slikt at det blir enklere å søke opp data som ligger i flere systemer. Videre kan man bruke tags for å spisse dokumentasjonen mer.

Det finnes ulike standarder for vokabular, både norske og internasjonale. Videre vil det innenfor de ulike tjenesteområdene finnes spesifikke standarder som brukes. Datakatalog ord og begreper skal så langt det lar seg gjøre følge Difi Dcat eller det som begrepsapparat som Stavanger kommune legger opp til. Der noe ikke er tilfredsstillende i standarden må Stavanger kommune komme opp med egne løsninger. Det skal legges til rette for at flere av disse standarden skal kunne brukes på tvers slik at man får en fleksibilitet som tilfredsstiller alle. Eier av data katalog vil være ansvarlig for å følge dette opp.

Et samarbeid med arkivenheten i Stavanger kommune skal alltid bli gjort nå man starter en dokumentasjon av kildesystem og data i datasjø. Det vil finnes flere overlappende arbeidsoppgaver og deling/samhandling skal gjøres der det er mulig. Spesielt av interesse her er at arkivenheten ofte kan ha gjort mye av dokumentasjonsarbeidet allerede, og det sparer da tid for dokumentering i datasjøen.

Arkivfunksjonen bruker i skrivende stund Decom applikasjonene fra Documaster. En framtidig løsning der man kan vurdere en integrasjon mellom Decom og Azure Data Catalog anbefales. Om det er fult integrert eller manuelt må bli vurder med tanke på hvem som er master.

15.4 Beste praksis i katalogisering av en kilde

Hver datakilde bør dokumenteres av en katalogeier (teknisk ekspertise) og en forretningseier (domene- og/eller systemekspertise).

Det bør være forretningseier som så blir tildelt ansvaret som eier for en gitt kilde med ansvar for å vedlikeholde kilden. Herunder elementer som brukervennlige navn, kommentarer, tags og andre metadata.

Ekstra viktig er det at kilder som er allment tilgjengelige er godt vedlikeholdt med tanke på gode metadata.

Kilder som allerede er markert for utfasing bør også dokumenteres nøye, så senere bruk av disse kildedataene ikke trenger å bety tungvint dokumentasjonsprosjekt.

Struktur og tydelig definisjon av hva som skal dokumenteres må etableres.

Et vokabular vil kunne ha flere dimensjoner.

	Beskrivelse	**Eksempel**
Stavanger kommune sine kilder og data	Kilder som eies og administrer under de ulike tjenesteområdene.	Stavanger kommune – Tjenesteområde – System – System Komponent/Funksjon
Prosesser for de ulike tjenesteområder	Under hvert tjenesteområde er de ulike prosesser. Her kan man knytte data til disse prosessene. Her kan man også tilrettelegge for bransjespesifikke standarder.	Støtte og Utvikling - HR – Ansatt – Lønn
GDPR	Struktur som beskriver data i forhold til GDPR standarden. Alle person- og sensitive data blir kategorisert slik at man senere kan dokumentere GDPR Vil kunne brukes opp mot datatilsynet.	GDPR – Data klassifisering • Personlige data • Særlige kategorier • Sensitive data • Ikke Personlig data GDPR – Formål • A • B

16 Navnekonvensjoner

Navnestandarder er et tema det er mange meninger om, og Azure slipper heller ikke unna at en gjør seg noen tanker om hvordan best navngi ressurser. Det er ikke alle ressurser som kan gis nytt navn så det er ingen god idé å starte med et midlertidig navn for å spare tid med en tanke om å finne et bedre navn senere.

En står forholdsvis fritt til å definere dette på egen hånd, men Azure har detaljerte regler for hvilke ressurser som støtter hvilken navngivning. Enkelte av disse restriksjonene er gitt ut fra eksterne faktorer som hvorvidt ressursen skal være adresserbar gjennom DNS, mens andre er styrt ut fra forskjellige behov internt i plattformen.

Det er dessverre ikke alt som er konsistent fra Microsoft sin side vedrørende støtte for store og små bokstaver, antall tegn, etc. Ikke benytt norske tegn (æ, ø, å) eller spesialtegn i navn.

Uten å gå i detalj på kravene for hver enkelt komponent i Azure kan det sis at det er to vesentlige artefakter å ha en navnestandard for: ressursgrupper og det som er felles for alle ressurser. Dette henger sammen med hvordan en segmenterer Azure-infrastrukturen. (Konto pr avdeling, ressursgruppe pr løsning, ressursgruppe pr miljø, etc.)

16.1 Ressursgrupper

De ulike miljøene skal ha hver sitt sky-abonnement. Men navnegivningen skal uansett gjenspeile hvilket abonnement gruppene tilhører. Så ressursgruppene skal kalles «Produksjon»/ «Test»/ «Utvikling»
I tillegg skal det vises hvilket prosjekt de tilhører, og hvilket lag de tilhører.

Og til sist, hvis det blir aktuelt å ha ressurser i flere regioner skal dette også inn i navnet. De ulike delene knyttes sammen med _ og hver komponent begynner med stor bokstav.

For eksempel Produksjon_Datasjø_Betjening_NE eller Produksjon_Datasjø_Transformasjon_Norge

16.2 Tjenester

Tjenester innen den enkelte ressursgruppe skal navngis etter følgende regler.

Ettersom ulike tjenester støtter ulik type navngivning (store og små bokstaver, skilletegn osv.), så kan det ikke legges til grunn en 100% korrekt navnestandard.

Derimot følger navnestandarden følgende mønster:

Der det er tillatt brukes <Beskrivende navn>_<Tjenestenavn>_<Miljø> - altså med _ mellom hver del, og stor første bokstav. For eksempel: Helseanalyse_CosDoc_Test.

Støttes ikke «-« eller «_» så skal alle deler slåes sammen til ett ord. For eksempel: HelseanalyseCosdocTest.

Der heller ikke store bokstaver tillates settes komponentene sammen med kun små bokstaver: «helseanalysecosdoctest».

Blir navn for lange og kompliserte kan for eksempel «+» benyttes som skille: «helse+analyse+cosdoc+test»

16.3 Språk

Språk i produkter og leveranser skal til enhver tid følge Stavanger kommune sin standard og være tilpasset målgruppen.

Det er en anbefaling at spesielt kode, kommentarer i kode, tabeller og navngiving av tjenester skal være på engelsk.

Ved bruk av Data Catalog vil man måtte dokumentere kilder som har engelsk som språk. Det må da besluttes om dette skal oversettes til norsk. Videre kan man vurdere å støtte flere språk slik at det i framtiden kan være søkbart for flere.

Vedlegg

17 Vedlegg A –Hva er data?

17.1 Informasjonsforvaltning / Informasjonsarkitektur

Definisjon fra Difi:

Informasjonsforvaltning betyr et helhetlig syn på aktiviteter, verktøy og andre tiltak for å sikre best mulig kvalitet, utnytting og sikring av informasjon i en virksomhet. Organiseringa av informasjonen skal være systematisk og henge sammen med virksomheten sine arbeidsprosesser.

For mer informasjon rundt dette kan man lese mer på DIFI sine sider.

Rammeverk for Datasjø og bruk av data i datasjø vil måtte underlegges samme regler som andre deler av informasjonsforvaltning i Stavanger kommune. Dette er

- Vurdering av tilganger til data
- Beskrivelse av data
- Valg av verktøy
    - Datakatalog brukes i dette rammeverket, men må følge andre systemer og standarder satt av Stavanger kommune
- Tilgjengelig gjøre beskrivelser
- Organisering, roller og prosesser i begrepsarbeidet

17.1.1 Datatyper

Dette er en veldig granulert liste, men vil gi en bedre forståelse av ulike typer av data når man analyserer kilder.

Data definisjon	Eksempel
Meta Data	Data om data. Ordlister generelt og for spesifikke fagområder.
Referanse Data	Kategoriseringer Vektenheter, landkoder, konverteringsregler, kalender
Master Data	Forretnings data Parter/Aktører/ Deltakere i transaksjoner Person, Avdeling, Sted, Kunde, Leverandør Hierarkier/relasjoner mellom master data objekter
Enterprise Struktur Data	Organisasjons strukturer, Geografiske strukturer, Ansvar Rapport dimensjoner
Transaksjons Data (Operasjonelle data)	Innkjøpsordre, faktura, kontrakt, sykemelding, målinger
Transaksjon data status	Framdrift/Status på transaksjons data

Sett fra et analyseståsted snakker man ofte om følgende inndeling av data

- Transaksjonsdata
    - I denne sammenheng er det data som brukes operasjonelt av forretningen og forvaltes i kildesystemer
- Analysedata
    - I denne sammenheng er det data som er hentet fra en eller flere kilder og som er transformert og tilrettelagt for beslutningsstøtte.
- Master Data
    - De sentrale enheter/byggeklossene for data

1.1.1 Klassifisering av data og informasjon

Data og Informasjon klassifiseres ofte på følgende måte.

- Åpen - Informasjon som *kan* eller *skal* være tilgjengelig for alle uten særskilte tilgangsrettigheter
- Intern - Dette er i utgangspunktet informasjon som ikke er åpen for alle.
- Fortrolig - Dette er informasjon som man er pålagt å begrense tilgangen til i lov, forskrift, avtaler, reglementer og annet regelverk. Dette tilsvarer graden fortrolig i den offentlige Beskyttelsesinstruksen
- Strengt fortrolig - Denne kategorien omfatter samme type informasjon som Fortrolig (rød), men der spesielle hensyn gjør at man ønsker å beskytte dataene ytterligere. Dette tilsvarer graden strengt fortrolig i den offentlige Beskyttelsesinstruksen

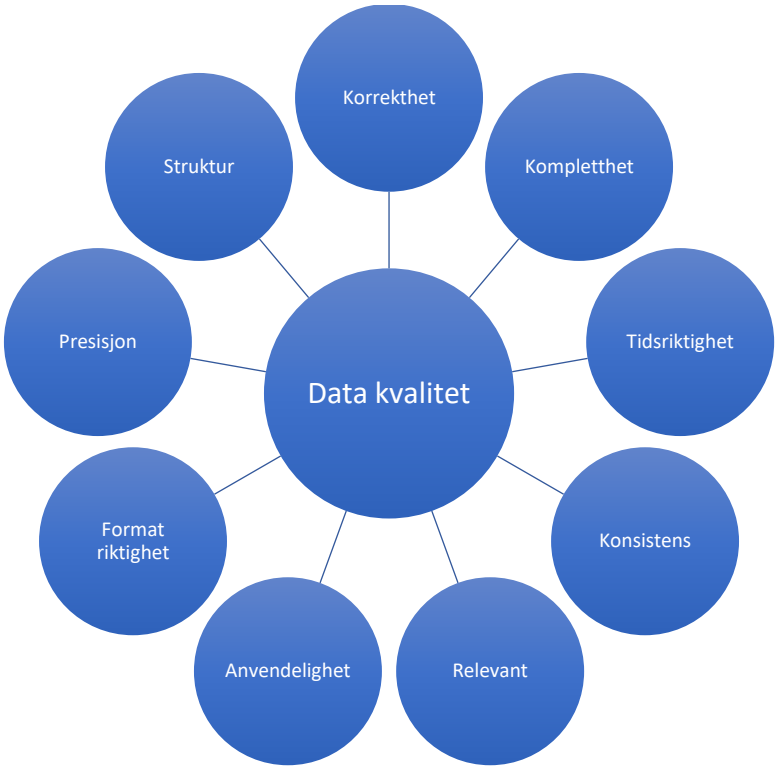
Difi forvalter åpne data via data.norge.no. Dette er data som er helt åpne for all og mottakere trenger ikke å være autentisert. Det er viktig at Stavanger kommune skiller på data som er helt åpne og data som kun er åpne for ansatte/interne.

17.1.2 Livssyklus

Data går igjennom en livssyklus. I denne konteksten er det antatt at livssyklusen til data ivaretas i kildesystemene og er i seg selv ikke dekket av dette dokumentet.

17.1.3 Datakvalitet

Datakvalitet er en av de viktigste områdene å ha kontroll på i et kildesystem, men ofte det som fører til mest problemer i et analyseprosjekt. Det er ofte der man ser problemer med manglende kontroll av datakvalitet.

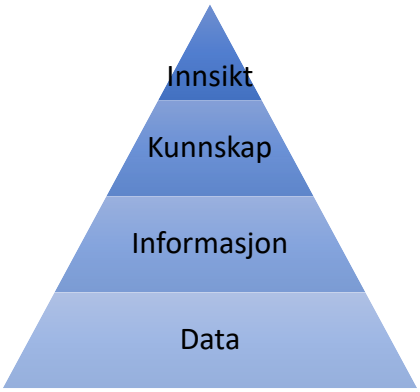


Det finnes ulike verktøy for å sette opp regler for datakatalog og som videre kan overvåke dataintegriteten.

17.2 Klassifisering av kunnskap – DIKW pyramiden

Data kan struktureres til informasjon. Data og informasjon kan formidles.

Kunnskap og innsikt må bygges opp individuelt eller i ei gruppe.



https://en.wikipedia.org/wiki/DIKW_pyramid

18 Vedlegg B –Lover og Regler

18.1 Personopplysingsloven / GDPR

I 2018 trådde den nye General Data Protection Regulation (GDPR eller personvernforordning) i kraft i EU/EØS-land, innlemmet i Norge av den nye personopplysningsloven. Loven omhandler beskyttelse av privat liv og regulerer innsamling og bruk av personopplysninger. Som hovedregel går personvernforordningen foran norsk lov ved konflikt, med mindre personvernforordningen åpner for nasjonale tilpasninger, såkalt særregler.

På bakgrunn av den nye loven er det enkelte personvernprinsipper Stavanger kommune er pliktige til å følge. Prinsippene gir på ulike måter uttrykk for at behandling av personopplysninger skal skje på en måte som i størst mulig grad sikrer forutsigbarhet og forholdsmessighet for enkeltmennesket.

Prinsippene er som følger:

Prinsipp	Forklaring
Lovlig, rettferdig og gjennomsiktig	Behandling av personopplysninger må være lovlig, dette betyr at det må finnes et behandlingsgrunnlag (rettslig grunnlag) for alle behandlinger av personopplysninger. Med rettferdighet menes det at opplysningene skal gjøres i respekt for de registrertes interesse og rimelige forventinger. Behandlinger skal være forståelig, og ikke foregå på en manipulerende eller skjult måte. Gjennomsiktig skal forstås som at behandlingen av personopplysningene skal være oversiktlige og forutsigbare for den som opplysningene gjelder
Formålsbegrensninger	Ethvert formål med behandling av personopplysninger skal identifiseres og beskrives presist, dette betyr at alle berørte skal ha samme forståelse av hva personopplysningene skal brukes til. Formålet skal i tillegg være i samsvar med øvrige etiske og rettslige samfunnsnormer.
Dataminimering	Mengden innsamlede personopplysninger skal ikke være større enn det som er nødvendig for å realisere formålet med innsamlingen. I de tilfeller der personopplysninger ikke er nødvendige, skal de heller ikke samles inn.
Riktighet	Personopplysninger som innsamles skal til enhver tid være korrekte. Dette betyr at behandlingsansvarlige er pliktige å sørge for å straks oppdatere eller slette de personopplysninger som ikke medfører riktighet ut ifra de formålene de er samlet inn for.
Lagringsbegrensninger	Når personopplysningene ikke lenger er nødvendige for det formålet de ble innhentet for, skal opplysningene slettes eller anonymiseres.
Integritet, konfidensialitet og tilgjengelighet	Integritet – Sikre mot utilsiktet/uautorisert endring eller sletting. Konfidensialitet – Sikre mot at uvedkommende får tilgang til opplysningene. Tilgjengelig – Det skal sikres at opplysningene alltid er tilgjengelig for det formålet de er tiltenkt.

	Den behandlingsansvarlige skal iverksette tiltak mot utilsiktede og/eller ulovlige ødeleggelser, tap og endringer av personopplysningene.
Ansvarlighet	En virksomhet må kunne dokumentere at tiltak er gjennomført for å etterleve personvernforordningen. Det skal jobbes proaktivt med etablering av nødvendige organisatoriske og tekniske tiltak, slik at regelverket etterleves til enhver tid.
Innebygd personvern	Personvern skal hensyn tas gjennom hele livssyklusen til dataene istedenfor å være noe som kommer i etterkant. Designet og innstillingene skal gjenspeile den mest personvernvennlige tilnærming. Det innebærer blant annet at Kommune har et ansvar for å sikre etterlevelse av ovenpå nevnt grunnleggende personvernprinsippene, i tillegg til å adoptere en proaktiv tilnærming til at disses etterlevelse. Man må spørre seg hvilken effekt skal iverksatt tiltak, prosesser og rutiner egentlig ha i praksis for å nå den ønskede beskyttelsesnivå? Det legges derfor vekt på hvordan overholdelse av prinsippene påvirker en helhetlig vurdering av personvern.

Datatilsynet har en rådgivende rolle hva gjelder personvern og personopplysninger. I tillegg fungerer Datatilsynet som er kontrollorgan som fører tilsyn med virksomheter for å avdekke samt rette opp manglende etterlevelse knyttet til personopplysningsloven.

18.1.1 Definisjoner i GDPR

Utvalgte definisjoner fra Personvernforordningen Art 4. -m

1) «**personopplysninger**» enhver opplysning om en identifisert eller identifiserbar fysisk person («den registrerte»); en identifiserbar fysisk person er en person som direkte eller indirekte kan identifiseres, særlig ved hjelp av en identifikator, f.eks. et navn, et identifikasjonsnummer, lokaliseringsopplysninger, en nettidentifikator eller ett eller flere elementer som er spesifikke for nevnte fysiske persons fysiske, fysiologiske, genetiske, psykiske, økonomiske, kulturelle eller sosiale identitet,

2) «**behandling**» enhver operasjon eller rekke av operasjoner som gjøres med personopplysninger, enten automatisert eller ikke, f.eks. innsamling, registrering, organisering, strukturering, lagring, tilpasning eller endring, gjenfinning, konsultering, bruk, utlevering ved overføring, spredning eller alle andre former for tilgjengeliggjøring, sammenstilling eller samkjøring, begrensning, sletting eller tilintetgjøring,

3) «**begrensning av behandling**» merking av lagrede personopplysninger med det som mål å begrense behandlingen av disse i framtiden,

4) «**profilering**» enhver form for automatisert behandling av personopplysninger som innebærer å bruke personopplysninger for å vurdere visse personlige aspekter knyttet til en fysisk person, særlig for å analysere eller forutsi aspekter som gjelder nevnte fysiske persons arbeidsprestasjoner, økonomiske situasjon, helse, personlige preferanser, interesser, pålitelighet, atferd, plassering eller bevegelser,

5) «**pseudonymisering**» behandling av personopplysninger på en slik måte at personopplysningene ikke lenger kan knyttes til en bestemt registrert uten bruk av tilleggsopplysninger, forutsatt at nevnte tilleggsopplysninger lagres atskilt og omfattes av tekniske og organisatoriske tiltak som sikrer at personopplysningene ikke kan knyttes til en identifisert eller identifiserbar fysisk person,

6) «**register**» enhver strukturert samling av personopplysninger som er tilgjengelig etter særlige kriterier, enten samlingen er plassert sentralt, er desentralisert eller spredt på et funksjonelt eller geografisk grunnlag,

7) «**behandlingsansvarlig**» en fysisk eller juridisk person, en offentlig myndighet, en institusjon eller ethvert annet organ som alene eller sammen med andre bestemmer formålet med behandlingen av

personopplysninger og hvilke midler som skal benyttes; når formålet med og midlene for behandlingen er fastsatt i unionsretten eller i medlemsstatenes nasjonale rett, kan den behandlingsansvarlige, eller de særlige kriteriene for utpeking av vedkommende, fastsettes i unionsretten eller i medlemsstatenes nasjonale rett,

8) «**databehandler**» en fysisk eller juridisk person, offentlig myndighet, institusjon eller ethvert annet organ som behandler personopplysninger på vegne av den behandlingsansvarlige,

9) «**mottaker**» en fysisk eller juridisk person, offentlig myndighet, institusjon eller ethvert annet organ som personopplysninger utleveres til, enten det dreier seg om en tredjepart eller ikke. Offentlige myndigheter som kan motta personopplysninger innenfor rammen av en særskilt undersøkelse i samsvar med unionsretten eller medlemsstatenes nasjonale rett, skal imidlertid ikke anses som mottakere; nevnte offentlige myndigheters behandling av slike opplysninger skal være i samsvar med gjeldende regler om vern av personopplysninger i henhold til formålet med behandlingen,

10) «**tredjepart**» enhver annen fysisk eller juridisk person, offentlig myndighet, institusjon eller ethvert annet organ enn den registrerte, den behandlingsansvarlige, databehandleren og de personer som under den behandlingsansvarlige eller databehandlerens direkte myndighet har fullmakt til å behandle personopplysninger,

11) «**samtykke**» fra den registrerte enhver frivillig, spesifikk, informert og utvetydig viljesytring fra den registrerte der vedkommende ved en erklæring eller en tydelig bekreftelse gir sitt samtykke til behandling av personopplysninger som gjelder vedkommende,

12) «**brudd på personopplysningssikkerheten**» et brudd på sikkerheten som fører til utilsiktet eller ulovlig tilintetgjøring, tap, endring, ulovlig spredning av eller tilgang til personopplysninger som er overført, lagret eller på annen måte behandlet,

13) «**genetiske opplysninger**» personopplysninger om en fysisk persons nedarvede eller ervervede genetiske egenskaper som gir unik informasjon om den aktuelle fysiske personens fysiologi eller helse, og som særlig er framkommet etter analysering av en biologisk prøve fra den aktuelle fysiske personen,

14) «**biometriske opplysninger**» personopplysninger som stammer fra en særskilt teknisk behandling knyttet til en fysisk persons fysiske, fysiologiske eller atferdsmessige egenskaper, og som muliggjør eller bekrefter en entydig identifikasjon av nevnte fysiske person, f.eks. ansiktsbilder eller fingeravtrykksopplysninger,

15) «**helseopplysninger**» personopplysninger om en fysisk persons fysiske eller psykiske helse, herunder om ytelse av helsetjenester, som gir informasjon om vedkommendes helsetilstand,

24) «**relevant og begrunnet innsigelse**» en innsigelse mot et utkast til avgjørelse om hvorvidt det foreligger en overtredelse av denne forordning eller om hvorvidt et planlagt tiltak som gjelder den behandlingsansvarlige eller databehandleren, er i samsvar med denne forordning, og som tydelig viser betydningen av risikoene som utkastet til avgjørelse utgjør med hensyn til de registrertes grunnleggende rettigheter og friheter og, dersom det er relevant, den frie flyten av personopplysninger i Unionen

18.1.2 Personvernkonsekvensvurdering DPIA

Personvernkonsekvensvurdering (DPIA) skal være en del av Kommunes risikostyring og inngå i dens internkontrollsystem.

Ved all behandling av personopplysninger må det avgjøres om innebygdpersonvern og de risikoreduserende tiltakene er tilstrekkelige, jf. personopplysningsloven artikkel 25. I denne vurderingen skal det tas hensyn til den tekniske utviklingen, gjennomføringskostnadene, behandlingens art, omfang, formål og sammenhengen den utføres i, samt risikoene av varierende sannsynlighets- og alvorlighetsgrad for fysiske personers rettigheter og friheter.

Disse vurderingene må dokumenteres og bygger grunnlaget til risikohåndteringsarbeid og ledelsens årlige gjennomgang. Når risikoen for de registrertes rettigheter og friheter antas som høy må behandlingsansvarlig i tillegg utføre en personvernkonsekvensvurdering (DPIA), jf. personvernforordningen artikkel 35. Målingen av risikonivået krever derfor at Kommune definer hva er akseptabel risiko, og når disse grensene overskrids. Det bør utføres en Pre DPIA (før personvernkonsekvensvurdering) på alle prosjekter for å dokumentere om behovet for en personvernkonsekvensvurdering (DPIA), er tilstede eller ikke. Hvis det er tvil om nødvendigheten av en DPIA skal personvernombudet i kommunen inkluderes i avgjørelsen.

Personvernkonsekvensvurdering (DPIA) er en prosess som skal beskrive og utrede hvordan behandlingen av personopplysninger påvirker personvern til de registrerte og veie behovet og hensiktsmessighet for videre tiltak for å minimere sannsynlighet for at uønskede henvendelser inntreffer eller for å minimere alvorlighetsgrad til konsekvenser for de registrertes liv, integritet, rettigheter, friheter, eiendom, omdømme og privatliv. Informasjon som er gitt til de registrerte og de registrertes forventninger er også viktig å ta i betraktning.

Ved identifikasjon av behov for risikoreduserende tiltak, skal Kommune involvere relevante interessenter i diskusjon, inkludert personvernombud. Godkjente tiltak skal deretter planlegges, implementeres og dokumenteres. Etter implementasjonen blir tiltakene testet for å avgjøre om de er hensiktsmessig i minimering av risikonivået. Hvis risikonivået fortsatt vurderes som for høyt, tas det en ny utredning av hvilke videre tiltak kan gjennomføres og prosessen gjentas. Dersom videre tiltak ikke reduserer risikoen til et tilfredsstillende nivå, må Kommune vurdere å ikke utføre behandlingen eller stoppe den (arvet behandling), eller gjennomføre en forhåndsdrøfting med Datatilsynet.

Personvernkonsekvensvurdering er et viktig verktøy for å forstå ansvarlighet, ettersom det hjelper den behandlingsansvarlige med å sikre samsvar med kravene i lovverket, samt å dokumentere at det overholdes. Dette arbeidet må forankres i ledelsen hos Kommune. Det er også Kommunes ledelse som må godkjenne risikovurderingene og tilstrekkeligheten av tiltakene som skal iverksettes for å håndtere konsekvenser og sannsynlighet for avvik som overskrider terskelen for akseptabel risiko.

Det anbefales at det brukes en standardisert mal, for å sikre harmonisering av intern praksis.

18.2 ISO og tilhørende referanser

ISO er en internasjonal standard som stiller krav til etablering, implementering, vedlikehold og kontinuerlig forbedring av et ledelsessystem for informasjonssikkerhet (https://www.standard.no/fagomrader/ikt/it-sikkerhet/isoiec-27001/). Til tross for at virksomheter ikke er pliktet til å følge standarden, velger mange organisasjoner likevel å implementere standardene eller blir sertifiseres i henhold til den.

Denne listen er ikke å anse som uttømmende, men gir eksempelvis punkter som krever kontinuerlig oppfølging i henhold til et forsvarlig informasjonsstyringssystem jf, ISO 27001:

Sikkerhetsmekanisme	**ISO 27001, Annex A**
Azure IAM	A.9.1.2 Access to networks and network services. A.9.4.1 Information access restriction
Azure Storage Service encryption (SSE)	A.10.1.1 Policy on the use of cryptographic controls.
Azure Key Vault	A.10.1.2. Key Management.
Hardware Security Module (HSM)	A.10.1.2. Key Management.
Microsoft Antimalware for Azure	A.12.2.1 Controls against malware
Azure security Center	A.12.4.1 Event logging
Azure Monitor	A.12.4.1 Event logging
Azure web application Firewall (WAF)	A.12.6.2 Restrictions on software installation

Krav/best praksis som beskrives i ISO 27001 kan deretter suppleres/kombineres med andre internasjonale standarder, som NIST-standarder, slik at de blir mer praksis-orientert. Her er det viktig å utpeke at implementering av disse standarder omfatter ikke bare løsningen, men også selv arbeidsmetoder og styring av informasjon i virksomhet.

18.2.1 ROS-analyse

Risikoarbeidet i en virksomhet er sentralt i etablering av et forsvarlig internkontrollsystem. Dette arbeidet bør gjennomføres i nær sammenheng og være i linje med IT-strategi og generelle strategiske mål til organisasjonen og skal gjenspeile hvordan virksomheten håndterer og kontrollerer risiko i den daglige. Risikoarbeidet er derfor et element som støtter prosesser, rutiner og selv drift, og sørger for at disse ikke utsetter virksomheten, informasjonsverdier eller tredjeparter, inkludert de registrerte, for uønskede hendelser.

I et slikt prosjekt, bør Kommune gjennomføre trussel- og risikovurderinger med jevne mellomrom for å avgjøre om det foreligger noen trussel/risiko eller om risikobildet har endret seg. Da utarbeides det av Kommunen en risikovurderingsplan som sikrer full dekning og tilrettelegger for at Kommunen kan ta informert beslutninger om hvordan det skal gås frem med vurderingen og avbøte risikoen.

Risikoer bør bli klassifisert og håndtert i henhold til dens sannsynlighet og alvorlighetsgrad i et objektivt og avtalt skåringssystem. Dette vil gi partene et realistisk grunnlag for vurdering av risikoene og sikre en felles forståelse for alle involverte interessenter av risiko som er til stede.

I nye personvernforordningen stilles det krav til at behandlingsansvarlige og databehandlere skal gjennomføre risikovurderinger. Risikovurdering relatert til *personopplysningssikkerhet* skal alltid gjennomføres før alle behandlinger og oppdateres regelmessig og ved endringer. Risikovurderingen skal identifisere områder som kan medføre utilsiktet eller uautorisert (ulovlig) tilgang, endring, sletting, tap eller utlevering av personopplysninger (art. 32 GDPR). Målet med vurderingen er å få kunnskap om hvilken risiko som eksisterer. Dette er nødvendig for å kunne iverksette tilstrekkelig risikoreduserende tiltak så man kan oppnå et akseptabelt sikkerhetsnivå ved behandlingen. Når det eksisterer risiko for de *registrertes rettigheter og friheter*, skall det i tillegg vurderes behovet for å gjennomføre en DPIA, som beskrevet ovenpå (se punkt 14.1.1).

19 Vedlegg C – Revisjon

19.1 Forvaltningsrammeverk

Et rammeverk for gjennomføring og forvaltning skal gi brukerne rammer og føringer for hvordan ledelse ønsker at prosesser og oppgaver skal løses, slik at de daglige oppgaver gjennomføres i henhold til lovkrav og interne retningslinjer og prosedyrer. Samtidig skal det muliggjøre at det verifiseres hvis retningslinjer, føringer og prosedyrer fungerer hensiktsmessig. Som andre deler av et vel fungerende internkontrollsystem, må forvaltningsrammeverket derfor også vurderes kontinuerlig. Disse kontrollfunksjoner bør nedfelles og er en integral del av kontrollerende dimensjon av internkontrollsystemet. Det er et krav at rammeverket har en eier. Denne eieren må invitere sentrale brukere til gjennomgang slik at rammeverket og dens regler kan vurderes og endres ved behov. Å etablere en tydelig HUKI-matrise og rapporteringslinjer er derfor et viktig verktøy for å sikre at alle interessenter er involvert i gjennomgangsprosessen og at prosessen er brukerrettet og støttet daglige funksjoner jf. tjenstlig behov. Rammeverkets styrker og svakheter vil bli avdekket ved de første implementasjonene det blir brukt i. Endringer og nye implementeringer bør ikke kunne gjennomføres uten forutgående godkjennelse av eieren og, muligens, relevante interessenter.

Eier til prosessen skal likevel vær varsom på hvordan endringer håndteres. Ved gjennomføring av store endringer, kan det skape usikkerhet i de allerede pågående prosjekter. Det kan også være utfordrende å kommunisere endringer til enkeltpersoner i prosjekter. Eventuelle endringer må tydelig kommuniseres til alle.

Unntak til ovenpå nevnt krav for godkjennelse er endringer som har som formål å håndtere sårbarheter eller avvik i tjenesten. I et slikt tilfelle vil de som håndterer disse hendelser informere eieren og relevante interessenter uten ugrunnet opphold. Alle endringer vil dokumenteres.

19.2 Sikkerhet

Ved nye implementeringer eller endringer i applikasjoner eller implementasjoner som bruker data i datasjøen, skal det gjennomføres en ROS-analyse jf. Punkt 0

Krav/best praksis som beskrives i ISO 27001 kan deretter suppleres/kombineres med andre internasjonale standarder, som NIST-standarder, slik at de blir mer praksis-orientert. Her er det viktig å utpeke at implementering av disse standarder omfatter ikke bare løsningen, men også selv arbeidsmetoder og styring av informasjon i virksomhet.

19.2.1 ROS-analyse

, før den nye implementasjonen eller endringer igangsettes eller det gis tilgang til data/data konsumeres.. Dette er jo spesielt viktig hvor data med strengere klassifiseringer er omfattet. Videre bør det vurderes behov for strengere kontroll av sikkerhet, og spesielt hvis endringen gjelder en integrasjon, f.eks. gjennom penetrasjonstesting. Hensynet er å unngå at sårbarheter med de nye implementasjonene/endringene kan utnyttes for å svekke sikkerhetsnivået og risikobildet.

Som en del av den ROS-analysen, skal det verifiseres behov for tilgang og brukerrettigheter i henhold til tjenstlig behov, slik at innstillingene som iverksettes i systemet etterlever innebygd personvern og innebygd sikkerhet prinsippene.